UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
ENGENHARIA DE COMPUTAÇÃO

GIONATTA MOCELLIN
LAURA WOBETO
THAYSE MARQUES SOLIS
WALDIR MARIN NETO

TABiR – TRAVA COM ABERTURA BIOMÉTRICA OU REMOTA

MONOGRAFIA

CURITIBA
2013

GIONATTA MOCELLIN
LAURA WOBETO
THAYSE MARQUES SOLIS
WALDIR MARIN NETO

TABiR – TRAVA COM ABERTURA BIOMÉTRICA OU REMOTA

Monografia apresentada, como forma parcial de avaliação, à disciplina Oficina de Integração 3 do curso de Engenharia de Computação do Campus Curitiba da Universidade Tecnológica Federal do Paraná.

Orientadores: Prof. Dr. João A. Fabro
Prof. Dr. Heitor S. Lopes

CURITIBA
2013

AGRADECIMENTOS

Certamente estes parágrafos não irão atender a todas as pessoas contribuíram para este projeto. Desde já pedimos desculpas àquelas que não estão presentes entre essas palavras, mas elas podem estar certas que fazem parte do nosso pensamento e gratidão.

Reverenciamos primeiramente os Professores João A. Fabro e Heitor S. Lopes pela orientação em cada fase do projeto, pelo empréstimo de materiais e também pelo apoio.

Agradecemos aos demais professores e colegas que de alguma forma ajudaram, dando ideias, sugerindo melhorias, acompanhando o desenvolvimento e dando suporte: Professor Ubiradir Mendes Pinto, Professor Percy Nohama, colegas Felipe Michels Fontoura, Michael Brochonski, Vinicius Martins e Maysa Ayame Takeuchi.

Nossos agradecimentos aos funcionários da Universidade Tecnológica Federal do Paraná pelo empréstimo de materiais e pela liberação do espaço para realização de testes.

Agradecemos aos professores da banca examinadora pela atenção e contribuição dedicadas a este trabalho.

Gostaríamos de deixar registrado também, nosso reconhecimento aos nossos pais, amigos e parentes que deram suporte emocional, apoiaram e entenderam nossa ausência durante o período de desenvolvimento do projeto, pois acreditamos que sem o apoio deles seria muito difícil vencer esse desafio.

RESUMO

Controle de acesso é um problema comum em diversos tipos de ambientes. Existem diversas soluções que implementam esta funcionalidade, que vão desde simples fechaduras até sistemas mais elaborados de controle biométrico. O projeto descrito é uma fechadura controlada por um sistema embarcado, que se baseia tanto na identificação da impressão digital quanto em comandos recebidos de algum dispositivo remoto para abri-la. O sistema é dividido em quatro módulos: o sistema embarcado (que inclui a fechadura), o sistema de comunicação, a estação base, e um aplicativo móvel. O sistema embarcado deve estar conectado à estação base para que o sistema funcione adequadamente. Todo usuário que for cadastrado na estação-base pode abrir a fechadura através da colocação de um de seus dedos sobre o sensor biométrico a ela conectado. Alternativamente, a fechadura pode ser aberta a partir de um aplicativo móvel que se comunica com a estação base. Sendo idealizado como um sistema de segurança, toda ação realizada no sistema embarcado, estação base ou aplicativo móvel é registrada, para permitir posterior auditoria.

ABSTRACT

Access control is a common concern in various environments. There are many ways to implement this, from simple locks to elaborate biometric systems. The project described here is a lock controlled from an embedded system based both on fingerprint recognition and on commands from a remote device. The system is divided in four modules: embedded system (which includes the lock itself), communication system, base station and mobile device. The embedded system must be connected to the base station so the system can properly function. Any user previously registered at the base station can open the door by putting their finger over the attached fingerprint sensor. It is also possible that the lock is opened remotely from a mobile device which communicates to the base station. The project intention is to function as a security system, so all actions are logged, and therefore it is possible to audit it.

LISTA DE FIGURAS

Figura 1 - Visão geral do projeto 17
Figura 2 - Diagrama de casos de uso da estação base 36
Figura 3 - Diagrama de classes da estação base 43
Figura 4 - Tela de login do software da estação-base. 44
Figura 5 - Aba de configurações do software da estação-base 45
Figura 6 - Tela de captura de impressões digitais. 46
Figura 7 - Aba de “CRUD” de usuários do software da estação-base 47
Figura 8 - Aba de “CRUD” para operadores do software da estação-base. 48
Figura 9 - Aba de registro de entrada e saída do software da estação-base. 49
Figura 10 - Aba de registro de eventos de operadores. 49
Figura 11 - Diagrama de casos de uso do sistema embarcado 50
Figura 12 - Diagrama de classes do firmware 53
Figura 13 - Máquina de estados do firmware para recebimento de mensagens .. 54
Figura 14 - Máquina de estados do firmware para funcionamento do sensor biométrico. 55
Figura 15 - Máquina de estados do firmware para monitoramento do botão 55
Figura 16 - Diagrama de blocos do hardware 57
Figura 17 - Esquema do optoacoplador 58
Figura 18 - Trava elétrica 61
Figura 19 - Interior da trava elétrica 61
Figura 20 - Esquema interno do LM555. 63
Figura 21 - Esquemático do circuito do microcontrolador 64
Figura 22 - Esquemático do circuito da trava. 64
Figura 23 - PCB do circuito do microcontrolador 65
Figura 24 - Parte superior da PCB do circuito da trava. 66
Figura 25 - Parte inferior da PCB do circuito da trava 66
Figura 26 - Diagrama de classes do celular 71
Figura 27 - Diagrama de caso de uso do celular 71
Figura 28 - Interface do aplicativo móvel 73
Figura 29 - Máquina de estados do aplicativo móvel. 76

Figura 30 - Máquina de estados da estação-base para o recebimento de requisição enviada pelo celular 77
Figura 31 - Máquina de estados da estação base para um envio de mensagem ao sistema embarcado 81
Figura 32 - Máquina de estados da estação base para o recebimento de mensagens do sistema embarcado..... 82
Figura 33 - Máquina de estados do sistema embarcado ao receber uma mensagem da estação-base 83
Figura 34 - Máquina de estados completa da estação base 102
Figura 35 - Máquina de estados completa do sistema embarcado 103

LISTA DE SIGLAS, ABREVIATURAS E ACRÔNIMOS

TI	Tecnologia da informação
LED	Light-emitting diode
CRUD	Create, Read, Update and Delete
DHCP	Dynamic Host Configuration Protocol
PCB	Printed Circuit Board

SUMÁRIO

1. INTRODUÇÃO 11
1.1. MOTIVAÇÃO 11
1.2. OBJETIVOS 12
1.2.1. Objetivos específicos 12
1.3. PREMISSAS 12
1.4. RESTRIÇÕES 13
1.5. TRABALHOS CORRELATOS 14
1.6. ESTRUTURA DO TRABALHO 15
2. PLANEJAMENTO 16
2.1. VISÃO GERAL 16
2.2. REQUISITOS 18
2.2.1. Requisitos da Estação Base 18
2.2.2. Requisitos do Sistema Embarcado 19
2.2.3. Requisitos do Celular 19
2.2.4. Requisitos da Comunicação Estação-base Sistema Embarcado ... 19
2.2.5. Requisitos da Comunicação Celular Estação-base 20
2.3. ALTERNATIVAS TECNOLÓGICAS 20
2.3.1. Estação-base 20
2.3.1.1. Linguagem de programação no computador 20
2.3.1.2. Comunicação com o sensor 22
2.3.2. Sistema embarcado 22
2.4. Microcontrolador 23
2.4.1.1. Linguagem de programação embarcada 25
2.4.1.2. Sensor biométrico 27
2.4.1.3. Trava elétrica 28
2.4.1.4. Bateria 28
2.4.2. Celular 29
2.4.3. Comunicação estação-base sistema embarcado 30
2.4.4. Comunicação celular estação-base 30
3. EXECUÇÃO 31
3.1. METODOLOGIA 31
3.2. CRONOGRAMA 31
3.3. ORÇAMENTO E GASTOS 34
3.4. PLANO DE RESPOSTA AOS RISCOS 34
4. ESTAÇÃO BASE 36
4.1. DIAGRAMAS 36
4.1.1. Diagrama de casos de uso 36
4.1.2. Descrição dos casos de uso 36
4.1.2.1. Caso de uso: Cadastrar usuário 36
4.1.2.2. Caso de uso: Remover usuário 37
4.1.2.3. Caso de uso: Consultar usuários 38
4.1.2.4. Caso de uso: Consultar log 39
4.1.2.5. Caso de uso: Atualizar cadastro 39
4.1.2.6. Caso de uso: Enviar dados 40
4.1.2.7. Caso de uso: Receber e armazenar dados no log 40
4.1.2.8. Caso de uso: Autorizar destravamento 41
4.1.2.9. Caso de uso: Receber requisição de destrancamento remota 41

4.1.2.10. Caso de uso: Validar login e senha 42
4.1.3. Diagrama de classes 43
4.1.4. Diagrama de estados 43
4.2. INTERFACE E USO DO SOFTWARE 44
5. SISTEMA EMBARCADO 50
5.1. FIRMWARE 50
5.1.1. Diagramas 50
5.1.1.1. Diagrama de casos de uso 50
5.1.1.2. Descrição dos casos de uso 50
5.1.1.3. Diagrama de classes 52
5.1.1.4. Diagrama de estados 53
5.1.2. Implementação 53
5.2. HARDWARE 56
5.2.1. Diagrama em blocos 56
5.2.2. Circuito do microcontrolador 57
5.2.2.1. Sensor 59
5.2.3. Circuito da trava 60
5.2.3.1. Trava elétrica 60
5.2.4. Circuito de alimentação 62
5.2.5. Diagramas elétricos 63
5.2.6. Projeto das PCBs 65
5.2.7. Lista de componentes para confecção da PCB completa 66
5.2.8. Guia de montagem 68
6. CELULAR 71
6.1. DIAGRAMAS 71
6.1.1. Diagrama de classes 71
6.1.2. Diagramas de casos de uso 71
6.1.3. Descrição do caso de uso 72
6.2. Caso de uso: Solicitar destrancamento 72
6.3. INTERFACE E USO DO SOFTWARE 72
7. PROTOCOLO DE COMUNICAÇÃO 74
7.1. COMUNICAÇÃO ENTRE ESTAÇÃO-BASE E CELULAR 74
7.2. COMUNICAÇÃO ENTRE ESTAÇÃO-BASE E SISTEMA EMBARCADO 78
8. RESULTADOS E CONCLUSÕES 84
8.1. ANÁLISE DO DESENVOLVIMENTO 84
8.2. INTEGRAÇÃO 85
8.3. TRABALHOS FUTUROS 86
9. REFERÊNCIAS 87
APÊNDICE A - CRONOGRAMAS PLANEJADO E EXECUTADO 88
APÊNDICE B - TABELAS DO PLANO DE RESPOSTAS A RISCOS 92
APÊNDICE C - MÁQUINA DE ESTADOS COMPLETA DA ESTAÇÃO BASE 102
APÊNDICE D - MÁQUINA DE ESTADOS COMPLETA DO FIRMWARE 103

1. INTRODUÇÃO

O projeto da Trava com Abertura Biométrica ou Remota - TABiR foi desenvolvido para a disciplina Oficina de Integração 3 do curso de Engenharia de Computação durante o segundo semestre letivo de 2012.

Os módulos requeridos para projeto são: uma estação-base, um sistema embarcado e um sistema de comunicação. A equipe optou por adicionar um quarto módulo: um aparelho celular que também se conecta com a estação-base. Na estação-base encontra-se o software responsável pela interface com o usuário. O sistema embarcado é constituído por um sistema microcontrolado, circuitos de alimentação e está conectado a uma trava elétrica.

1.1. MOTIVAÇÃO

O interesse da equipe por diferentes meios de autenticação usados em sistemas de segurança, além da possibilidade de controle à distância motivou a escolher um projeto que pudesse solucionar um problema comum: o acesso a ambientes restritos.

Existem ambientes em empresas e lojas que não podem ser acessados por qualquer funcionário. Um exemplo seria uma sala de servidores em uma empresa de TI: o acesso de alguém não autorizado e com más intenções poderia comprometer o trabalho de todos.

Uma possível solução para esse problema seria limitar a entrada através de identificação biométrica, fazendo os devidos registros. Além disso, seria interessante habilitar o acesso ao ambiente por algum dispositivo com acesso a Internet, caso algum usuário não cadastrado necessite acesso e o gerente não esteja presente para liberar ou realizar o cadastro deste usuário.

O projeto inclui um sistema embarcado, formado por um microcontrolador, um sensor e um botão, que se comunica com uma estação-base (computador) via transmissão de dados cabeada Ethernet, usando protocolo TCP/IP. Há também comunicação entre um dispositivo remoto (celular) e a estação-base via *sockets*. Logo, são atendidos os requisitos propostos para o projeto da Disciplina.

1.2. OBJETIVOS

O objetivo geral deste projeto é desenvolver um sistema que possibilita acesso à uma sala restrita por meio de dados biométricos ou remotamente.

1.2.1. Objetivos específicos

- Permitir a entrada em um ambiente restrito pela aquisição de impressão digital junto à porta;
- Permitir a liberação da porta por um comando remoto enviado pelo celular a qualquer distância da estação-base (desde que haja uma rede móvel disponível);
- Permitir a saída do ambiente protegido pelo pressionamento de um botão;
- Registrar em um *log* na estação-base as entradas (com os dados de autenticação) e as saídas;
- Cadastrar, atualizar e remover dados de um usuário deste sistema na estação-base;
- Registrar em um *log* as modificações que um operador deste sistema fizer.

1.3. PREMISSAS

- A estação-base estará sempre ligada e conectada à Internet;
- O sistema embarcado deve ter uma fonte de alimentação;
- O celular deve possuir acesso à Internet;
- O desbloqueio da trava realizado via celular será feito assim que a estação base receber o comando e repassar ao sistema embarcado;
- O cadastro de um novo usuário será feito na estação-base, com o apoio de um sensor idêntico ao do sistema embarcado e digitação dos dados complementares através de um teclado;
- A estação-base estará conectada ao sistema embarcado através de uma rede local cabeada. Foi adotada a opção de conexão cabeada Ethernet pelo

fato da implementação de uma rede sem fio depender de um roteador de longo alcance, que aumentaria os custos do projeto;

- O registro do destravamento da porta feito pelo lado de fora inclui data e hora de entrada, nome do funcionário e seu ID no banco de dados;
- As informações armazenadas no destravamento feito pelo lado de dentro da sala são: data e hora de saída.
- Os registros na estação-base serão feitos em um log de dados sequencial;
- Há um botão no lado de dentro da sala protegida, próximo à tranca;
- Há um sensor biométrico do lado de fora da sala protegida, próximo à tranca;
- Caso a entrada seja permitida, um indicador luminoso (LED) verde será aceso, caso contrário, um indicador luminoso (LED) vermelho acenderá, indicando que o usuário não recebeu autorização para entrar.
- Sempre que a estação-base se conectar com o sistema embarcado, ocorrerá a sincronização das impressões digitais presentes nos arquivos binários da estação-base.

1.4. RESTRIÇÕES

O tempo disponível para realização do projeto impacta na colocação de restrições sobre seu funcionamento. São elas:

- O travamento da tranca é feito mecanicamente apenas; é necessário puxar a porta com as mãos para que ela tranque.
- A identificação biométrica é realizada apenas do lado de fora da sala protegida.
- O destravamento pelo lado de dentro da porta é feito através de um botão.
- O computador (da estação-base) deve ter pelo menos uma porta acessível pela rede pública.
- Não será possível o destravamento via celular se a estação-base não estiver operando ou não possuir acesso à internet.
- Não será possível o destravamento via celular se este não possuir acesso à Internet.
- Não há comunicação direta entre o celular e a tranca.

- Quando o destravamento ocorrer pelo celular, não será realizada identificação biométrica através do sensor localizado do lado de fora da sala.
- Durante o funcionamento do sistema, os cabos de rede não devem ser desconectados, uma vez que a obtenção dos endereços IP da estação-base e do sistema embarcado é feita via DHCP.
- O número máximo de usuários que podem ser cadastrados para reconhecimento biométrico é 40, uma vez que o sensor armazena no máximo 160 impressões digitais. Por decisão de projeto, são armazenadas 4 impressões digitais, de 4 dedos diferentes, para cada usuário.
- O ID inicial para cadastro das impressões digitais de um usuário na estação-base deve ser um número múltiplo de quatro e menor que 157.
- O ID de cada impressão digital é único.
- Usuários só poderão ser cadastrados se o sensor biométrico foi localizado pelo software da estação-base.
- Usuários só poderão ser removidos se houver conexão entre estação-base e sistema embarcado.

1.5. TRABALHOS CORRELATOS

O trabalho “Projeto de uma trava elétrica microprocessada” [1] desenvolvido por Júnior Freire Coelho e Vânio da Maia faz o uso de um microcontrolador PIC para desenvolver um sistema de segurança de baixo custo, cuja interface homem-máquina consiste num teclado e um display LCD. As senhas para destrancar a trava são constituídas de seis números de zero a nove e o projeto permite até três tentativas para destrancar a porta. Caso a terceira tentativa falhe, um alarme sonoro é disparado. O projeto não faz uso de uma estação-base que se comunica com o sistema embarcado, nem permite o destravamento remoto.

O projeto "Sistema de controle do fluxo de funcionários utilizando leitor biométrico" [2] foi desenvolvido por estudantes do Instituto de Estudos Superiores da Amazônia e consiste em um simulador de um sistema de controle que compara a digital de um funcionário com a de um banco de dados

e libera ou não o acesso deste à determinado local. O sensor biométrico utilizado é conectado diretamente ao computador via USB, onde está rodando o software de registro e acesso. Não é utilizado nenhum sistema mecânico (ex. trava ou catraca) e nenhum dispositivo móvel. É utilizado apenas um Arduino como simulador desse sistema mecânico, onde há dois LEDs (verde e vermelho) que acendem nos casos de liberação e bloqueio de acesso, respectivamente.

1.6. ESTRUTURA DO TRABALHO

Na próxima seção será apresentado o planejamento do projeto contendo uma visão geral do que foi proposto, e os requisitos que levaram a um estudo de alternativas tecnológicas.

Na seção 3 (Execução), é apresentada a metodologia utilizada no desenvolvimento, o cronograma planejado e o realizado, os gastos previstos e os efetivos, assim como o plano de resposta aos riscos desenvolvido para tratar qualquer situação que tenha atrapalhado o desenvolvimento do projeto.

Na seção "Estação base" são expostos os diagramas e descrições de casos de uso, diagrama de classes e de estados e a descrição da interface e uso do software desenvolvido para a estação-base.

Em "Sistema Embarcado" são mostrados os diagramas relativos ao firmware e hardware assim como detalhes da implementação do software embarcado, explicação dos circuitos, diagramas elétricos, lista de componentes e guia de montagem do hardware.

Na seção 6 (Celular) são exibidos os diagramas de classe, caso de uso e a descrição da interface e uso do software desenvolvido para o aplicativo móvel.

A seguir, na seção 7, apresentam-se as informações relativas ao protocolo de comunicação tanto entre a estação-base e o celular como entre a estação-base e o sistema embarcado.

Por fim, a seção 8 expõe os resultados e conclusões sobre o projeto envolvendo uma análise do desevolvimento, da integração com outras matérias do curso de Engenharia de Computação e ideias para melhorar e ampliar o escopo do projeto em trabalhos futuros.

2. PLANEJAMENTO

2.1. VISÃO GERAL

A Figura 1 mostra uma visão geral do sistema proposto, composto por uma estação-base (um computador), um sistema embarcado, um celular e um sistema de comunicação (composto pelo Protocolo TCP/IP por meio da rede Ethernet entre a estação-base e o sistema embarcado e por meio da Internet entre o celular e a estação-base).

O objetivo deste projeto é impedir o acesso de pessoas não autorizadas a um ambiente restrito, produzindo um sistema de desbloqueio de uma tranca elétrica através da aquisição de dados de uma impressão digital (por meio de um sensor biométrico). O envio destes dados, do sistema embarcado até a estação base, será feito através de transmissão de dados cabeada usando protocolo Ethernet.

A liberação do acesso também poderá ser feita a partir de um dispositivo remoto (celular) via Internet. Além disso, há um botão que habilita a abertura da porta pelo lado de dentro. As entradas autenticadas e saídas serão registradas em um *log* na estação-base.

O destravamento externo será feito por um sensor biométrico, integrante do sistema embarcado, localizado do lado externo da sala, próximo à tranca, ou por um comando enviado remotamente via celular. O destravamento interno será feito através de um botão localizado do lado interno da sala, próximo à tranca, que provocará a abertura e enviará uma notificação à estação-base. O bloqueio da tranca será feito mecanicamente da mesma forma que grande parte dos sistemas de segurança comerciais.

O software da estação-base possibilitará o cadastro de novos usuários autorizados a entrar assim como a remoção usuários já registrados, fará o registro das entradas e saídas através dos dados recebidos do sistema embarcado e armazenará esses dados em um log.

O cadastro de novos usuários será feito através da aquisição das digitais dos dedos polegar e indicador tanto da mão esquerda como da direita no sensor da estação-base, além da inserção de informações como nome, CPF e cargo dentro da empresa. Não há fotos dos funcionários no banco de dados.

A entrada na sala consiste na análise da impressão adquirida pelo sensor. Habilita-se a entrada, ou não, de acordo com a comparação dos dados recebidos e dos presentes em um banco de digitais armazenado no sistema embarcado. O usuário recebe uma indicação se a entrada foi permitida ou não através de um sinal luminoso (acendimento de um LED verde no caso de acesso permitido ou de um LED vermelho no caso de acesso negado). No caso da saída da sala, um botão será pressionado e os dados trocados com a estação-base servirão apenas para registro de uma saída e não para autenticação do usuário que deixou o ambiente, uma vez que não há controle do agente que acionou o botão.

Os dados armazenados durante uma abertura por fora são: data e hora de entrada, nome do funcionário e seu ID no banco de dados. As informações armazenadas na abertura por dentro são: data e hora de saída.

O software do celular será desenvolvido de forma a possibilitar o destrancamento da porta à distância. Não há limite para essa distância, desde que o celular e a estação-base estejam conectados à Internet. Há uma senha que o usuário do celular deve digitar para que o aplicativo libere a opção de destrancamento remoto da porta. O administrador do sistema pode autorizar usuários a destravar a porta pelo celular, criando para eles uma senha no momento de seu cadastro.

Figura 1 - Visão geral do projeto

2.2. REQUISITOS

A seguir serão apresentados os requisitos do projeto. Os requisitos são condições do projeto necessárias para alcançar um objetivo, satisfazer um padrão ou função.

Os requisitos foram divididos em cinco categorias: estação-base, sistema embarcado, celular, comunicação estação-base sistema embarcado e comunicação estação-base celular.

2.2.1. Requisitos da Estação Base

- A estação-base deverá ter suporte e possuir mouse, teclado e monitor.
- O computador da estação-base deverá ter sistema operacional Windows XP ou superior.
- A estação-base deverá ter porta USB (universal serial bus).
- O software da estação-base deverá manter um registro das aberturas da porta pelo lado de fora e das aberturas pelo lado de dentro. Esse registro será representado por texto.
- A estação-base deverá ter acesso à Internet, com uma porta pública disponível para uso pelo software.
- A estação-base deverá ter um sensor biométrico igual ao do sistema embarcado.
- A estação-base deverá ser capaz de comunicar-se com o sistema embarcado através de *sockets* TCP.
- A estação-base deverá ser capaz de aceitar conexões provenientes do celular, usando *sockets*.
- O software da estação-base deverá permitir cadastro, remoção e consultas de usuários.
- O *log* salvo na estação-base deverá incluir data e hora de entrada, nome do funcionário e seu ID no banco de dados no caso de um destravamento feito pelo lado de fora da sala.
- O *log* salvo na estação-base deverá incluir data e hora de saída no caso de um destravamento feito pelo lado de dentro da sala.

- O *log* salvo na estação-base será armazenado em formato de texto.

2.2.2. Requisitos do Sistema Embarcado

- O sistema embarcado deverá acompanhar uma tranca elétrica.
- O sistema embarcado deverá ter um sensor de digital posicionado do lado de fora da tranca.
- O sistema embarcado deverá abrir pelo lado de fora se um usuário cadastrado posicionar os mesmos dedos que foram cadastrados sobre o sensor de digital.
- O sistema embarcado deverá ter um mecanismo de abertura pelo lado de dentro (botão).
- O sistema embarcado deverá armazenar os dados de impressões cadastradas mesmo depois de desligado.
- O sistema embarcado deverá ter uma fonte alternativa de alimentação, além da forma principal, que alimente o circuito temporariamente no caso de falta de energia.
- O sistema embarcado deverá ter indicadores luminosos para mostrar se o mesmo está ligado e para sinalizar autorização/negação da entrada.

2.2.3. Requisitos do Celular

- O celular deverá ter acesso a Internet não cabeada (Wi-Fi ou 3G).
- O celular deve ter um sistema para o qual seja possível desenvolver software gratuitamente.
- O celular deverá comunicar-se com a estação-base quando necessário.
- O software do celular possibilitará o desbloqueio da tranca pela inserção de um login e uma senha.

2.2.4. Requisitos da Comunicação Estação-base Sistema Embarcado

- A comunicação entre estação-base e sistema embarcado deverá ser bidirecional.

- A comunicação entre estação-base e sistema embarcado deverá ser via rede local cabeada.

2.2.5. Requisitos da Comunicação Celular Estação-base

- A comunicação entre estação-base e celular deverá ser bidirecional.
- A comunicação entre estação-base e celular deverá ser via Internet.

2.3. ALTERNATIVAS TECNOLÓGICAS

Durante a análise de opções tecnológicas são apontadas alternativas viáveis e exploradas novas ferramentas para solucionar um problema em especial. Além disso, é discutido um plano alternativo no caso de falha da primeira escolha, minimizando os riscos do projeto.

Apresentam-se a seguir as alternativas tecnológicas em cinco grupos: estação-base, sistema embarcado, celular, comunicação estação-base sistema embarcado e comunicação estação-base celular.

2.3.1. Estação-base

A estação-base é um computador onde será instalado o software utilizado para cadastrar/remover usuários, guardar informações sobre os usuários cadastrados e armazenar horários de abertura da porta.

A tecnologia estudada nesta etapa envolve a linguagem de desenvolvimento do software que fará a interface entre o usuário e as informações sobre os usuários, assim como as alternativas para comunicação com o sensor no qual as impressões digitais são adquiridas durante o cadastro de novos usuários.

2.3.1.1. Linguagem de programação no computador

A linguagem de programação deve facilitar o trabalho do programador em termos de desenvolvimento e manutenção do código, assim como resultar em um software de fácil utilização pelo usuário. Algumas alternativas seriam as

linguagens C, C++ ou Java. Outras alternativas, poderiam ser Python ou C#, mas foram descartadas pelo fato de nenhum membro da equipe conhecê-las em profundidade e não ser possível estudá-las no intervalo de tempo proposto. Apresenta-se um resumo da comparação dos critérios na Tabela 1.

Tabela 1 - Comparação das características relativas às linguagens de programação

Linguagem	***C, C++***	***Java***
Portável?	Não.	Sim.
Requerimentos adicionais?	Não.	Computador tem de ter uma JVM instalada.
Eficiência	Alta	Média (interpretado) ou alta (JIT)
Facilidade de manutenção do código	Menor.	Maior.
API gráfica	Não, mas é possível usar bibliotecas como Allegro, GTK, entre outras, ou a API do sistema operacional de destino.	Sim.
Suporte a comunicação serial	Não nativa, mas possível utilizando bibliotecas externas, como a Boost (Boost Software License), ou a API do sistema operacional de destino.	Não nativa, mas possível utilizando biblioteca RxTx (LGPL).
Gerenciamento de memória	Gerenciamento explícito.	Possui *garbage collector*.
Documentação	Não possui um mecanismo de documentação *inline* padrão, mas é possível usar Doxygen em C++.	Padrão de documentação Javadoc
Familiaridade dos membros da equipe.	Média	Alta

A programação de interfaces em linguagens C ou C++ é bem mais complexa e dependente de bibliotecas externas. O código não é facilmente portável como o é em Java, e a manutenção é mais trabalhosa, o gerenciamento de memória tem que ser explícito, sua documentação não é padronizada.

A linguagem Java apresenta-se como uma alternativa para a confecção da interface gráfica. Java apresenta uma API simples e bem-documentada para a criação de interfaces, tornando a programação mais simples. Java também tem a vantagem de ser facilmente portável; os arquivos não são compilados para linguagem de máquina, e sim para um código intermediário (bytecode) que executa sobre uma máquina virtual, sendo possível usar um mesmo executável em Java em qualquer sistema para o qual haja uma máquina virtual compatível. Optou-se, portanto, pelo uso de Java na programação da interface com o usuário.

2.3.1.2. Comunicação com o sensor

O sensor escolhido utiliza comunicação serial e requer alimentação de 3,6 a 5,0 V. Logo, um conversor USB para serial com a alimentação adequada é necessário. Foi encontrado apenas um item que satisfazia às necessidades conforme a Tabela 2.

Tabela 2 - Comparação das características relativas ao conversor USB para serial

Item	***Conversor Usb P/ Serial FTDI Chip***
Tensão de alimentação	3,3 ou 5 V
Sinais	RXD, TXD, RTS e CTS
Preço (R$)	46

2.3.2. Sistema embarcado

O sistema embarcado é o módulo do projeto que envolve o microcontrolador, o sensor biométrico que ficará junto à porta, a trava elétrica e o botão. A escolha correta de cada uma das partes que compõe esse módulo é fundamental para o bom funcionamento do projeto.

A análise desta seção está dividida em: microcontrolador, linguagem de programação embarcada, sensor biométrico e trava elétrica.

2.4. *Microcontrolador*

O microcontrolador é o dispositivo principal do sistema embarcado. Ele também é o responsável pelo controle dos outros dispositivos empregados no projeto como: o LED que acende quando a entrada é liberada, o circuito que destrava a tranca, os dispositivos de comunicação, dentre outros.

Foi feita uma pesquisa, incluindo a disponibilidade, facilidade de encontrar e custo de microcontroladores e foram selecionados três (AT89C5131, LPC1768 e ATmega328). Para analisar qual traria um melhor custo-beneficio para o projeto, contando não somente custo financeiro, mas também o quanto cada opção irá impactar no possível tempo de aprendizado do mesmo, a Tabela 3, abaixo, apresenta uma comparação entre eles.

Tabela 3 - Comparação das características relativas aos microcontroladores

Microcontrolador	***AT89C5131***(Datasheet ATMEL AT89C5131) ***(família MCS-51)***	***LPC1768***(Datasheet NXP LPC1768) ***(família Cortex-M3)***	***ATmega328***(Datasheet ATmega 328) ***(família megaAVR)***
Clock	48MHz (modo X1) 24MHz (modo X2) (externo)	100MHz	16 MHz
Controlador Ethernet	Não	Sim	Não, mas alguns kits Arduino já vêm com controlador ethernet Wiznet.
Portas de I/O de uso geral	34 pinos de I/O	26 pinos de I/O (no kit da mbed, mas o controlador permite uso de até 70 pinos de GPIO)	23 pinos de I/O
UART	1	4	1 (inclui biblioteca para fazer por software em outros pinos)
Biblioteca para ethernet	Versão antiga da µIP (não mantida) para usar com controlador ENC28J60 (requer RAM externa).	Biblioteca acompanha a IDE.	Biblioteca acompanha a IDE.

Memória ISP	32 kB	512 kB	32 kB
Memória RAM interna	256 B (+ 1 kB[1])	64 kB	2 kB
Capacidade Máxima de Endereçamento Externo	64 kB	Não tem suporte direto a endereçamento externo	Não tem suporte direto a endereçamento externo
Memória RAM Total Máxima	64 kB + 256 B	64 kB	2 kB
Kit de interfaceamento e gravação	P51USB	mbed NXP LPC1768 prototyping board	Arduino Ethernet[2]
Ferramenta de desenvolvimento	Keil uVision 4 (linguagens Assembly/C)	mbed Compiler (linguagens C/C++)	Arduino IDE (linguagem C)
Custo total do kit	Aprox. R$ 100,00	Aprox. R$ 150,00 (sem imposto) ou R$ 240,00 (com imposto)	Aprox. R$ 255,00 (Arduino Ethernet)

Como o microcontrolador deverá controlar o destrancamento da trava, a emissão de luz do LED indicador de permissão, o recebimento dos comandos de abertura da trava e efetuar comunicação com a estação-base, de forma "simultânea" do ponto de vista da percepção humana, foi estimado (mantendo uma boa margem de segurança) que o microcontrolador deverá possuir no mínimo uma capacidade de processamento de 0,1 MIPS, ou seja, 100 mil instruções por segundo. Nesse aspecto, todos os microcontroladores apresentados na tabela 3 servem com bastante folga, com destaque para o LPC1768, que possui um *clock* de 100 MHz.

Outra importante característica levantada refere-se ao controlador Ethernet já no processador, uma vez que sem o mesmo é necessário comprar um módulo externo e eventualmente implementar uma pilha TCP ou UDP, ou usar uma pilha pronta, o que leva muito tempo e é trabalhoso. Nesse quesito o At89C5131 está descartado, uma vez que não possui controlador Ethernet no

[1] O At89C5131 tem memória expandida on-chip, de forma que os primeiros 1 kB que seriam da memória externa podem ser acessados a partir da RAM interna.
[2] Alternativamente é possível combinar Arduino Uno/Mega + Arduino Ethernet Shield, obtendo resultado semelhante, mas com maior custo.

kit. As outras duas alternativas, LPC1768 e ATmega328 são viáveis uma vez que já trazem esse recurso em seus kits.

Outro recurso importante é referente ao protocolo. Ficou definido que seria utilizado protocolo TCP/IP para a comunicação entre o sistema embarcado e a estacao-base. O LPC1768 ja contem as pilhas do protocolo implementadas, economizando assim tempo no decorrer do projeto para a realização de outras implementações.

Fora isso, o controlador da família Cortex-M3, em comparação com o ATmega328, também possui maior número de portas de entrada e saída de dados para que possam ser acessados, lidos e controlados os periféricos utilizados no sistema embarcado. E em relação ao preço também o LPC1768 é uma melhor alternativa que o ATmega328.

Analisados esses aspectos a equipe chegou à conclusão que o microcontrolador que deverá ser utilizado para o projeto é o LPC1768, pois além dele cumprir todas as características necessárias para que o mesmo seja utilizado no projeto também é o que possui melhor custo/benefício.

2.4.1.1. Linguagem de programação embarcada

Para a programação em sistemas embarcados, algumas opções de linguagens, são Assembly e C. Outras alternativas, como a linguagem Ada, não serão levadas em conta devido ao fato de serem incomuns ou pouco conhecidas. Todas as linguagens apresentadas permitem usar o hardware para as funcionalidades necessárias (comunicação serial, entrada e saída de dados, etc). A comparação dos critérios respectivos às linguagens apresenta-se resumida na Tabela 4.

Tabela 4 - Comparação das características relativas às linguagens de programação embarcadas

Linguagem	***Assembly***	***C***	***C++***
Dependência da arquitetura	Alta	Baixa	Baixa
Tipo	Baixo nível.	Procedural.	Procedural e orientada a objetos
Nível de abstração	Baixo	Médio	Alto
Complexidade	Alta	Alta/Média	Média

Familiaridade	Média (MCS-51)	Média	Alta
	Nenhuma (Cortex M-3)		
	Nenhuma (megaAVR)		
Documentação disponível	Sim	Sim	Sim

A linguagem Assembly apresenta um conjunto de mnemônicos (conhecidos como "opcodes"), equivalentes a instruções em linguagem de máquina. Assembly possibilita a criação de programas muito eficientes, por se programar diretamente em instruções do processador, mas às vezes a complexidade torna-se muito grande. Portanto, a dificuldade de programação faz com que o trabalho demande mais tempo, e que haja menos confiabilidade no software resultante. Também é necessário considerar que a linguagem Assembly é diferente para cada família de microcontroladores, pois seus conjuntos de instruções são diferentes. Dessa forma, ao optar-se pelo uso de Assembly, dependendo da opção de microcontrolador adotada, pode ser possível que seja necessário aprender o Assembly daquela arquitetura. Atualmente, os membros da equipe têm familiaridade apenas com o Assembly da arquitetura Intel MCS-51.

A linguagem C e seus dialetos apresentam a facilidade de terem uma sintaxe mais amigável, permitindo programação em mais alto nível que aquela realizada em Assembly, já que é possível fazer um programa mais independente da arquitetura. Além disso, a maioria dos compiladores permite a adição de trechos de código em Assembly dentro de um programa em linguagem C. Portanto, a linguagem C é geralmente uma opção melhor que a linguagem Assembly.

Em comparação com a linguagem C++ a linguagem C apresenta algumas desvantagens, são elas: o nível de abstração mais baixo (não existem os conceitos de orientação a objetos, como classes, objetos e métodos) e o fato de o grupo ter menos familiaridade com ela. Algumas bibliotecas que acompanham a IDE da mbed são feitas em C++.

A linguagem C++ foi desenvolvida com base na linguagem C, mas não é propriamente uma extensão dela, já que um código C pode precisar de adaptação para ser utilizado em C++. Ainda assim, para a maioria dos usos simples, C++ pode ser vista como uma extensão de C. Dessa forma, a

linguagem C++ acaba sendo uma opção melhor, visto que um código feito em C geralmente pode ser utilizado em um programa feito em C++, mas o contrário é normalmente impossível. Dessa forma, é possível que as bibliotecas da mbed só sejam utilizáveis através de código C++.

Após ponderar as vantagens e desvantagens das opções tecnológicas para a programação do software embarcado, optou-se pela programação em linguagem C++.

2.4.1.2. Sensor biométrico

O sensor biométrico é o periférico do sistema embarcado responsável pela aquisição das impressões digitais, armazenamento e é pelos dados que ele coleta que o sistema irá permitir ou não a abertura da tranca.

A Tabela 5 mostra alguns dos sensores pesquisados assim como algumas características dos mesmos.

Tabela 5 - Comparação das características relativas aos sensores biométricos

Item	***Leitor Biométrico Bioprint***	***Leitor Biométrico USB APC*** (APC Touch Biometric Pod Password Manager)	***Leitor Biométrico All-in-one ZFM-20 (Adafruit)***(Fingerprint sensor)	***Leitor biométrico Suprema SFM3520***(Datasheet SFM3520-OP)
Captura imagem	Sim	Sim	Sim	Sim.
Armazena banco de digitais?	Sim, até 10 impressões	Sim, até 20 impressões	Sim, até 162 impressões	Sim, até 9000 impressões.
Identifica digitais?	Sim.	Sim	Sim	Sim.
Interface de comunicação	USB + driver (Windows)	USB + driver (Windows 98-XP e MAC OS)	Serial (RS-232)	Serial (RS232, RS422 ou RS485)

Software	Proprietário	Proprietário	Exemplo em código aberto (BSD) para Arduino	Nenhum, apenas especificação.
Preço médio (R$)	80,00	102,00	105,00	R$ 170,00

Tendo em vista o aspecto de que já há software de exemplo de conexão com um microcontrolador e a quantidade de impressões que podem ser armazenadas, o sensor mais adequado para o projeto é o ZFM-20, que é apenas 3 reais mais caro que o leitor APC, mas que possui características melhores. Na verdade, nenhum sensor que usa protocolo USB com driver proprietário é adequado para uso no projeto. O sensor SFM3520 foi descartado por não ter software de exemplo, e devido a seu preço elevado (70% mais caro que o ZFM-20).

2.4.1.3. Trava elétrica

O destravamento da trava elétrica será acionado pelo microcontrolador no sistema embarcado. É importante que o mecanismo envolvido não seja muito complexo e que o preço seja o menor possível. As travas pesquisadas possuem preços e características muito parecidas, logo, o critério para compra foi a reputação do vendedor e o frete mais barato.

A trava escolhida é do fabricante Morelle e é destrancada com a aplicação de 12V,. mecanismo simples que facilita o projeto, e o custo é de 64 reais.

2.4.1.4. Bateria

A bateria será responsável pela alimentação do circuito caso a alimentação da rede falhe. As baterias pesquisadas foram as apresentadas na Tabela 6.

Tabela 6 - Comparação das características relativas às baterias

Item	***Planet Battery***	***Unipower Up1213***	***Unipower Up1270***
Tensão	12 V	12 V	12 V
Carga	1,3 Ah	1,3 Ah	7 Ah
Flutuação	13,5 V / 13,8 V	13,5 V / 13,8 V	13,5 V / 13,80 V

Corrente inicial	0,39 A Máx	0,52 A Máx.	2,1 A Máx.
Preço médio (R$)	36,70	47,50	48,00

A bateria escolhida é a Unipower Up1270, uma vez que não possui preço tão mais elevado que as demais e sua corrente máxima é maior.

2.4.2. Celular

O celular, como meio de destravamento remoto, possui como principais requisitos ter acesso à internet e apoio à uma plataforma de desenvolvimento gratuita. Também é importante que a linguagem de programação utilizada no software para celular seja conhecida pelos membros da equipe, devido ao prazo de execução do projeto.

A seguir é mostrado na Tabela 7 um levantamento de vantagens/desvantagens em celulares de diferentes plataformas/sistemas operacionais.

Tabela 7 - Comparação das características relativas aos sistemas operacionais

	Android	***iOS (iPhone)***	***Windows Phone***
Linguagem de programação	Java (nativamente, usando a Dalvik), C# (usando Mono for Android).	Objective-C	.NET (C#, VB.NET, etc)
IDE gratuita?	Sim (Eclipse + ADT)	Sim (XCode)	Sim (Visual Studio Express)
Desenvolvimento gratuito?	Sim.	Não. Preço por desenvolvedor de US$ 99,00/ano.	Não. Preço por desenvolvedor de US$ 99,00/ano.
Distribuição gratuita?	Não. Taxa única de US$ 25,00, mais 30% da receita com a venda.	Não. Cobra-se 30% da receita com a venda.	Não. Cobra-se 30% da receita com a venda.

A linguagem Java facilita o trabalho pelos mesmos motivos expostos na seção de escolha da linguagem a ser usada no software da estação-base. É importante notar que a única opção que possibilita desenvolvimento gratuito é o

Android. Por esses motivos, escolheu-se trabalhar com um celular da plataforma Android, cujo desenvolvimento de software pode ser feito sem custo adicional com IDE e licença. Em especial, a maior vantagem da plataforma Android é o fato de a programação ser em linguagem Java, já conhecida da equipe.

2.4.3. Comunicação estação-base sistema embarcado

A comunicação entre a estação-base e o sistema embarcado é o que possibilitará a esses dois módulos trocar informações para que a entrada seja liberada e os logs de acesso sejam gravados. Será feita via cabo Ethernet. Conforme a seção 2.2, o microcontrolador terá integrado um controlador Ethernet, enquanto a estação-base deve ter uma interface de rede. Ambos devem estar ligados a um switch ou um roteador. Nesse caso o único custo adicional é com um conector RJ45 com isolamento magnético (“conector RJ45 com transformador”), para a conexão do cabo Ethernet junto ao controlador. Esse item custa aproximadamente 20 reais.

2.4.4. Comunicação celular estação-base

A comunicação entre a estação-base e o celular (figura 1) será feita por meio de *sockets*. O roteador que liga o computador (da estação-base) ao celular deve ter uma porta de acesso público aberta que o conecta à Internet. O computador ira escutar através dessa porta, aguardando uma possível conexão feita pelo celular. Esse mecanismo independe de hardware adicional (considera-se que o roteador já faz parte da infra-estrutura básica disponível), logo não há custo a ser acrescentado ao projeto proveniente da comunicação entre a estação-base e o celular.

3. EXECUÇÃO

3.1. METODOLOGIA

Para o desenvolvimento do projeto foi escolhido um modelo de processo iterativo e incremental (com varias iterações no tempo), gerando novas versões a cada release. O risco de implementar o software inteiro de uma só vez é maior do que ir planejando e desenvolvendo aos poucos; o retrabalho de uma correção do software inteiro é muito maior do que de partes mais simples.

As vantagens do processo incremental e iterativo são:

- possibilidade de avaliar mais cedo os riscos e pontos críticos do projeto, e identificar medidas para os eliminar ou controlar;
- existe sempre algo para mostrar ao cliente/sponsor (a última iteração);
- menor tempo de desenvolvimento do projeto como um todo, porque a equipe trabalha de maneira mais eficiente quando pretende alcançar resultados de escopo pequeno e claro;
- baseia-se na participação e uma boa comunicação entre desenvolvedores;
- ao fim de cada iteração pode-se ter um feedback para acompanhar como está o projeto; mesmo se não estiver de acordo com o proposto, ainda há tempo para mudanças;
- os testes em cada iteração são mais simples.

As desvantagens desse modelo são:

- dificuldade em prever as durações dos ciclos, afinal o número de ciclos e sua temporização depende do avanço do projeto;
- problemas que aparecem durante o desenvolvimento podem atrasar um ciclo.

3.2. CRONOGRAMA

A Tabela 13 apresenta o cronograma cumprido nos *Deliverables* durante o período de desenvolvimento do projeto. Todas as etapas foram cumpridas conforme planejado.

Os cronogramas completos de atividades, tanto o planejado quando o executado, podem ser visualizados no Apêndice A. As linhas destacadas em vermelho representam atrasos e as em verde, cumprimento dos prazos. Os atrasos na elaboração dos manuais deve-se ao fato de que a equipe preferiu progredir com o desenvolvimento para documentar um volume maior de informação. O atraso no final do cronograma foi devido ao fato de ter sido dada uma reserva de tempo para verificação de tudo o que foi feito, realização de mais testes e dedicação às demais disciplinas no final do semestre.

Tabela 8 - Descrição dos deliverables

Deliverable	**Descrição do artefato**	**Auxiliar do gerente**

13/03/2013	**Primeiro protótipo do software da estação-base :**
	• Diagramas UML (casos de uso, classe, sequência) do software da estação-base
	• Versão inicial da interface gráfica.
	  ○ Apresentará a janela de cadastro de usuários
	  ○ Apresentará a janela onde será mostrado o log de entrada e saída.
	  ○ Alguns botões ainda não são funcionais.
	  ○ Ainda não haverá comunicação com o celular
	
	Primeiro protótipo do software do celular:
	• Diagramas UML (casos de uso, classe, sequência, entidade-relacionamento) do software do celular
	• Interface com o usuário
	Ainda não haverá comunicação com a estação-base
		Waldir

27/03/2013	**Implementação do banco de dados** • Opções de cadastro, remoção e consulta de usuários. **Primeiros testes da comunicação entre estação-base e celular.** • A estação-base recebe os dados de login e senha, compara com os presentes no banco de dados e	Gionatta

	envia ao celular um sinal de comando aceito ou não. • Ainda não haverá retransmissão para o sistema embarcado do sinal de liberação ou bloqueio do acesso **Primeiros testes da comunicação entre estação-base e sistema embarcado** • estação-base -> sistema embarcado: envio de um único bit • sistema embarcado -> estação-base: envio de um único bit **Sistema embarcado:** • **Conexão do microcontrolador com a trava elétrica** ○ Ainda não haverá destravamento por biometria	
10/04/2013	**Mais testes de comunicação entre estação-base e sistema embarcado** • estação-base -> sistema embarcado: envio dos dados da impressão digital adquirida durante um cadastro **Segundo protótipo do software da estação-base** • Implementação do cadastro por via biométrica **Sistema embarcado:** • Adição do botão ao sistema embarcado, com sua função implementada	Laura
24/04/2013	**Comunicação entre estação-base e sistema embarcado** • sistema embarcado -> estação-base: reconhecimento de impressões e envio dos dados (dependendo da digital estar ou não armazenada no sensor) para atualização do log na estação-base. **Terceiro protótipo do software da estação-base** • Tratamento dos dados recebidos pelo embarcado: atualização do log de acesso.	Gionatta

3.3. ORÇAMENTO E GASTOS

Abaixo, na Tabela 9, é apresentada a lista de componentes com os custos planejados e os custos efetivos. Os preços estimados foram levantados baseando-se nos itens escolhidos na Análise de Opções Tecnológicas, considerando os que melhor atendiam aos requisitos do projeto e cujo custo fosse mais baixo.

Tabela 9 - Lista de componentes e seus preços estimados e efetivamente gastos

Item	**Quantidade**	**Preço orçado (R$)**	**Preço real (R$)**
Conversor USB/Serial	1	46,00	46,27
Microcontrolador LPC1768	1	240,00	158,00
Sensores biométricos	2	210,00	213,46
Trava	1	64,00	63,80
Bateria	1	48,00	39,60
Cabo Ethernet	1	10,00	6,00
Conector RJ45 com isolamento	1	20,00	23,50
Placa de circuito impresso + componentes (com reposição de peças)	1	210,00	140,12
Fontes de alimentação	2	30,00	60
TOTAL		878,00	750,75

Nota-se que foi gasto 85,5% do previsto. Tal diferença deve-se ao fato do microcontrolador não ter sido taxado pela Aduana e pelo fato da equipe ter planejado um gasto de R$80,00 de reserva para o caso de dano/perda/extravio de alguma peça de maior valor.

3.4. PLANO DE RESPOSTA AOS RISCOS

Riscos estão presentes em qualquer projeto. Para evitar surpresas e saber de antemão quais atitudes seriam tomadas em casos de ocorrência de algum risco, a equipe elaborou um Plano de Resposta aos Riscos presente no Apêndice B.

Ocorreram três dos riscos previstos: Subestimar a complexidade de implementação do software ou hardware, Problemas com o microcontrolador, sensor biométrico e/ou demais componentes, Não cumprimento dos prazos estabelecidos pelo gerente. O primeiro ocorreu com o planejamento e desenvolvimento do hardware, que tomou um pouco mais de tempo do que o previsto, levando à ocorrência do terceiro risco; o prazo das atividades internas da equipe não foi cumprido. Por último houve um problema mecânico com a trava, mas que foi solucionado com o apoio do Prof. Heitor S. Lopes.

4. ESTAÇÃO BASE

4.1. DIAGRAMAS

Esta seção apresenta alguns diagramas relativos à estação-base. São apresentados apenas os componentes da modelagem que a equipe julgou relevantes para o planejamento e desenvolvimento do software.

4.1.1. Diagrama de casos de uso

Na Figura 2 é apresentado o diagrama de casos de uso da estação base. Para uma melhor visualização, recomenda-se que esta figura seja consultada no CD-ROM que acompanha esta monografia.

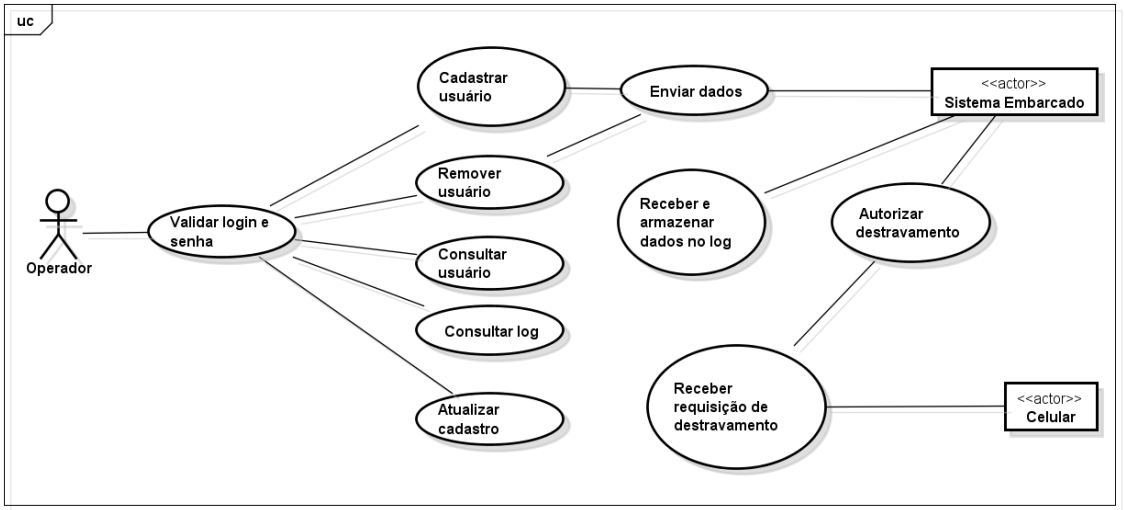

Figura 2 - Diagrama de casos de uso da estação base

4.1.2. Descrição dos casos de uso

4.1.2.1. Caso de uso: Cadastrar usuário

Ator Principal: Operador do software
Ator de Suporte: Usuário
Ator de bastidor: Sistema embarcado
Descrição: Executado quando deseja-se cadastrar um usuário

Pós-condições:

- Usuário cadastrado

Fluxo Básico:

1. Operador do software seleciona a opção de cadastro de usuários no software.
2. O sistema exibe a tela pedindo informações do usuário a ser cadastrado.
3. O operador preenche estas informações.
4. O operador confirma estas informações.
5. O Sistema verifica a validade das informações conforme as Regras de negócio. (a)
6. O Sistema pede a impressão digital do usuário.
7. O usuário coloca sua digital no sensor ligado à estação-base.
8. O Sistema confere os metadados da digital lida. (b)
9. O Sistema cadastra o usuário.
10. O Sistema manda os metadados do usuário cadastrado para o Sistema embarcado.

Fluxo Alternativo:

- A qualquer momento o operador cancela a operação: O Sistema desconsidera as informações e encerra o caso de uso.

a. Informações incorretas

a.1. Retorna ao fluxo básico 2.

b. Falha na leitura de dados

b.1. Exibe uma tela de informação sobre a falha na leitura.

b.2. Retorna ao fluxo básico 6.

Regras de negócio:

1. Todos os campos são obrigatórios.

4.1.2.2. Caso de uso: Remover usuário

Ator Principal: Operador do software

Descrição: Executado quando deseja-se remover um usuário

Pós-condições:

- Usuário removido.

Fluxo Básico:

1. Operador do software seleciona a opção de remoção de usuários no software.
2. O sistema exibe a tela pedindo informações do usuário a ser removido.
3. O operador preenche estas informações.
4. O operador confirma estas informações.
5. O Sistema verifica a existência do usuário no banco de dados conforme as Regras de negócio. (a)
6. O operador seleciona o usuário a ser removido.
7. O Sistema remove o usuário.

Fluxo Alternativo:

- A qualquer momento o operador cancela a operação: O Sistema desconsidera as informações e encerra o caso de uso.

a.Inexistência do usuário

a.1. Exibe uma mensagem de inexistência do usuário.

a.2. Retorna ao fluxo básico 2.

Regras de negócio:

1. O campo CPF deve ser preenchido.

4.1.2.3. Caso de uso: Consultar usuários

Ator Principal: Operador do software

Descrição: Executado quando se deseja consultar usuários

- Operador tem permissão para consultar

Pós-condições:

- Usuário(s) encontrado(s) ou não

Fluxo Básico:

1. Operador do software seleciona a opção de consulta de usuários no software.
2. O sistema exibe a tela pedindo informações do usuário a ser consultado.
3. O operador preenche estas informações.
4. O operador confirma estas informações.
5. O Sistema compara as informações com as presentes no banco de dados conforme Regras de negócio. (a)
6. O Sistema mostra uma tela com a resposta da consulta.

Fluxo Alternativo:

- A qualquer momento o operador cancela a operação: O Sistema desconsidera as informações e encerra o caso de uso.

Regras de negócio:

1. Todos os campos são opcionais.

4.1.2.4. Caso de uso: Consultar log

Ator Principal: Operador do software

Descrição: Executado quando se deseja consultar o log de entrada e saída

Fluxo Básico:

1. Operador do software seleciona a opção de consulta do log no software.
2. O sistema exibe a tela do log.

4.1.2.5. Caso de uso: Atualizar cadastro

Ator Principal: Operador do software

Descrição: Executado quando deseja-se atualizar um cadastro

Pré-condições:

- Usuário cadastrado

Pós-condições:

- Cadastro atualizado

Fluxo Básico:

1. Operador do software seleciona a opção de atualização de cadastro no software.
2. O sistema exibe a tela para escolha do cadastro a ser atualizado.
3. O operador preenche as informações.
4. O operador confirma estas informações.
5. O Sistema confirma a atualização.

Fluxo Alternativo:

- A qualquer momento o operador cancela a operação: O Sistema desconsidera as informações e encerra o caso de uso.

4.1.2.6. Caso de uso: Enviar dados

Ator Principal: Operador do software

Descrição: Executado quando operação de cadastro e remoção de usuário finalizada e consolidada.

Pré-condições:

- Usuário cadastrado;
- Usuário removido;

Pós-condições:

- Cadastro atualizado no sistema embarcado;

Fluxo Básico:

1. Operador do software cadastra ou remove usuário.
2. Operador confirma a ação anterior.
3. Estação-base envia dados para sistema embarcado.
4. Sistema embarcado atualiza informações armazenadas no sensor.

Fluxo Alternativo:

- A qualquer momento o operador cancela a operação de remover ou cadastrar: O Sistema desconsidera as informações;

4.1.2.7. Caso de uso: Receber e armazenar dados no log

Ator Principal: Sistema embarcado

Descrição: Executado quando um destravamento é realizado diretamente no sistema embarcado (tanto de entrada quanto de saída)

Pré-condições:

- Houve um destravamento

Pós-condições:

- Informações armazenadas no log

Fluxo Básico:

1. O Sistema recebe a informação do sistema embarcado se foi uma abertura por dentro ou por fora. (a, b)

Fluxo Alternativo:

a. Abertura por dentro (botão)

a.1. O Sistema armazena a data e hora do destravamento no log.

a.2. Encerra o caso de uso.

b. Abertura por fora (biometria)

b.1. O Sistema armazena data, hora e informações do usuário que realizou o destravamento no log.

b.2. Encerra o caso de uso.

4.1.2.8. Caso de uso: Autorizar destravamento

Ator Principal: Sistema embarcado

Descrição: Executado quando a estação-base recebe requisição de destravamento remota.

Pré-condições:

- Houve uma solicitação de destravamento remota;

Pós-condições:

- Destravamento da trava;

Fluxo Básico:

1. Recebe requerimento de abertura de porta verificado;
2. Envia solicitação de abertura de porta para sistema embarcado;
3. Sistema embarcado retorna a estação-base informando que houve abertura de porta ;
4. Estação-base envia retorno ao celular informando que a porta foi aberta;

Fluxo Alternativo:

- Rompimento da comunicação entre o sistema embarcado e a estação-base;
- Solicitação expira o tempo para a conexão;
- Retorna informando timeout da operação para estação-base;
- Envia informações de falha na comunicação com o sistema embarcado para celular;

4.1.2.9. Caso de uso: Receber requisição de destrancamento remota

Descrição: Executado quando a estação-base recebe uma requisição de destrancamento remota.

Pós-condições:

- Autorização de destravamento;

Fluxo Básico:

1. O Sistema recebe uma requisição de destravamento.
2. O Sistema compara os dados de login e senha recebidos com os presentes no banco de dados. (a)
3. O Sistema envia mensagem de destravamento ao sistema embarcado.
4. O Sistema envia mensagem de volta ao celular de autorização do destravamento.
5. O Sistema armazena no log as informações do destravamento.

Fluxo Alternativo:

a.Informações incorretas

a.1. O Sistema envia mensagem de rejeição ao celular.

a.2. Encerra caso de uso.

4.1.2.10. Caso de uso: Validar login e senha

Ator Principal : Operador

Descrição: Executado ao iniciar o software da estacao-base para dar acesso as funcionalidades do sistema;

Pós-condições:

- Acesso as funcionalidades do sistema;

Fluxo Básico:

1. O operador inicia o software;
2. Insere seu login e sua senha;
3. Os dados são verificados;
4. Operador tem acesso ao sistema;

Fluxo Alternativo:

a.Informações incorretas

a.1. O Sistema envia mensagem de rejeição a tela inicial do programa;

a.2. Encerra caso de uso.

4.1.3. Diagrama de classes

Na Figura 3 é apresentado o diagrama de classes da estação base. Para uma melhor visualização, recomenda-se que esta figura seja consultada no CD-ROM que acompanha esta monografia.

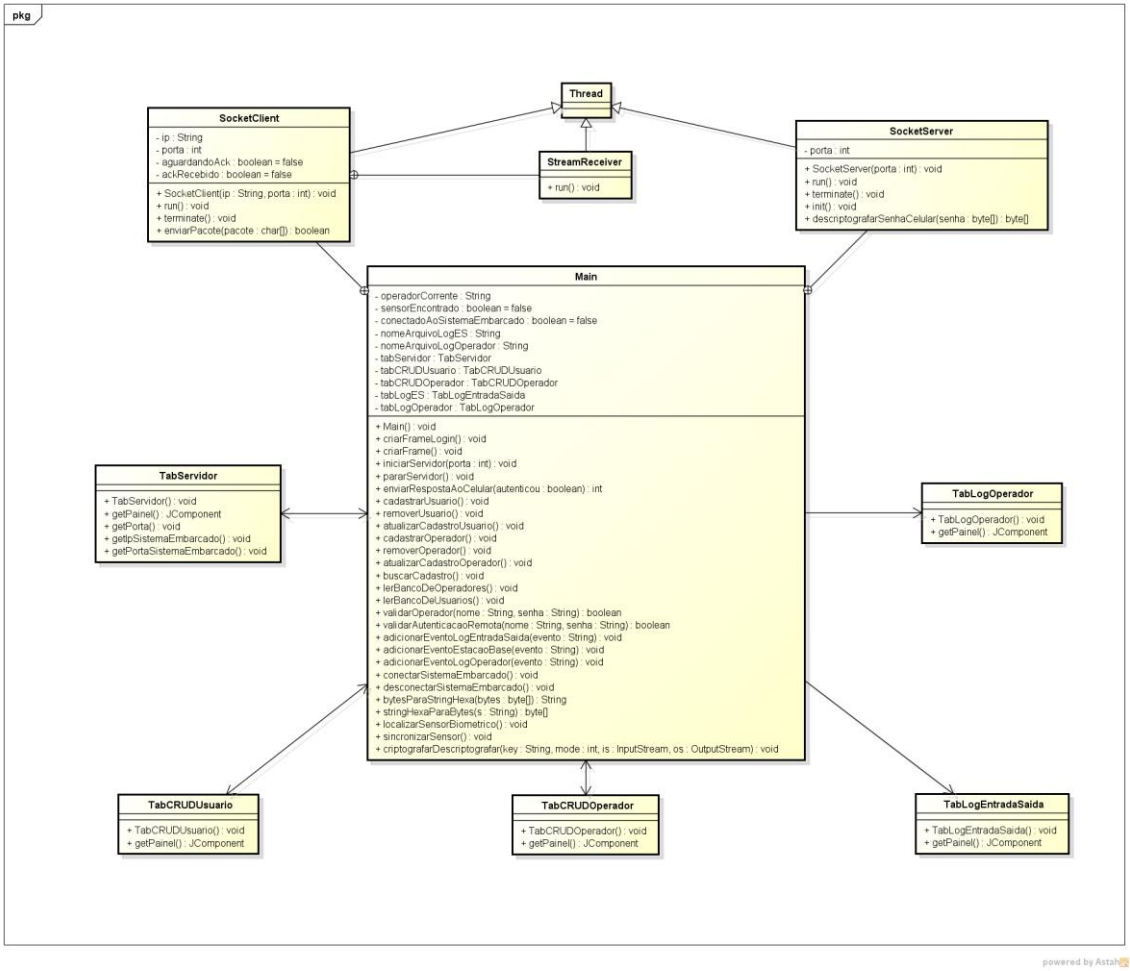
pkg
Thread
SocketClient
- ip : String
- porta : int
- aguardandoAck : boolean = false
- ackRecebido : boolean = false
+ SocketClient(ip : String, porta : int) : void
+ run() : void
+ terminate() : void
+ enviarPacote(pacote : char[]) : boolean
StreamReceiver
+ run() : void
SocketServer
- porta : int
+ SocketServer(porta : int) : void
+ run() : void
+ terminste() : void
+ init() : void
+ descriptografarSenhaCelular(senha : byte[]) : byte[]
Main
- operadorCorrente : String
- sensorEncontrado : boolean = false
- conectadoAoSistemaEmbarcado : boolean = false
- nomeArquivoLogES : String
- nomeArquivoLogOperador : String
- tabServidor : TabServidor
- tabCRUDUsuario : TabCRUDUsuario
- tabCRUDOperador : TabCRUDOperador
- tabLogES : TabLogEntradaSaida
- tabLogOperador : TabLogOperador
+ Main() : void
+ criarFrameLogin() : void
+ criarFrame() : void
+ iniciarServidor(porta : int) : void
+ pararServidor() : void
+ enviarRespostaAoCelular(autenticou : boolean) : int
+ cadastrarUsuario() : void
+ removerUsuario() : void
+ atualizarCadastroUsuario() : void
+ cadastrarOperador() : void
+ removerOperador() : void
+ atualizarCadastroOperador() : void
+ buscarCadastro() : void
+ lerBancoDeOperadores() : void
+ lerBancoDeUsuarios() : void
+ validarOperador(nome : String, senha : String) : boolean
+ validarAutenticacaoRemota(nome : String, senha : String) : boolean
+ adicionarEventoLogEntradaSaida(evento : String) : void
+ adicionarEventoEstacaoBase(evento : String) : void
+ adicionarEventoLogOperador(evento : String) : void
+ conectarSistemaEmbarcado() : void
+ desconectarSistemaEmbarcado() : void
+ bytesParaStringHexa(bytes : byte[]) : String
+ stringHexaParaBytes(s : String) : byte[]
+ localizarSensorBiometrico() : void
+ sincronizarSensor() : void
+ criptografarDescriptografar(key : String, mode : int, is : InputStream, os : OutputStream) : void
TabServidor
+ TabServidor() : void
+ getPainel() : JComponent
+ getPorta() : void
+ gettpSistemaEmbarcado() : void
+ getPortaSistemaEmbarcado() : void
TabLogOperador
+ TabLogOperador() : void
+ getPainel() : JComponent
TabCRUDUsuario
+ TabCRUDUsuario() : void
+ getPainel() : JComponent
TabCRUDOperador
+ TabCRUDOperador() : void
+ getPainel() : JComponent
TabLogEntradaSaida
+ TabLogEntradaSaida() : void
+ getPainel() : JComponent
powered by Astah

Figura 3 - Diagrama de classes da estação base

4.1.4. Diagrama de estados

O diagrama de estados da estação base está presente no Apêndice C. Como o diagrama é muito grande, recomenda-se que sua visualização seja feita através da figura original contida no CD-ROM que acompanha esta monografia.

4.2. INTERFACE E USO DO SOFTWARE

O software da estação-base foi desenvolvido em Java, conforme definido na etapa do Plano de Projeto. Foi utilizado o ambiente de desenvolvimento Eclipse para a sua implementação durante todo o projeto.

O software consiste de uma interface gráfica, com suas principais funções separadas por abas.

Para o reconhecimento de dispositivos serial, foi utilizada a biblioteca “RxTx”[3] para Java, disponível tanto para sistemas 32 bits quanto 64 bits.

Primeiramente, deve-se logar como um operador para que o software possa ser de fato iniciado para sua utilização, conforme mostrado na Figura 4.

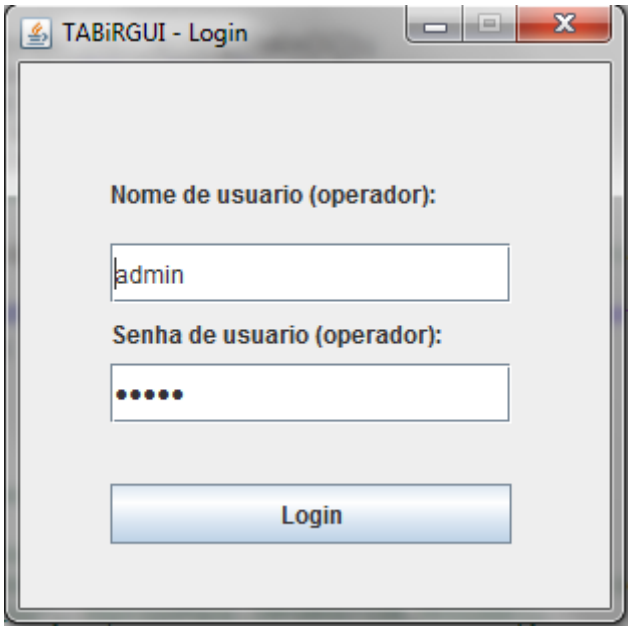

Figura 4 - Tela de login do software da estação-base.

A primeira aba, mostrada na Figura 5 possibilita que o operador realize operações de abertura de porta para habilitar o recebimento de requisições de destravamento do celular, via Internet; conexão ou desconexão com o sistema embarcado utilizado no projeto, sendo necessário saber o seu IP e a porta no momento da conexão para que a ela possa ser estabelecida; localizar o sensor biométrico, conectado à uma das portas USB da estação-base através de um driver USB-Serial, visto que o sensor possui conexão serial apenas, utilizado no cadastro de novos usuários.

Caso ocorra uma ativação da conexão com o sistema embarcado, inicia-se o processo de sincronização de dados entre a estação-base e o sistema embarcado, onde a estação-base envia as minúcias de cada impressão digital presente em seu banco de dados ao sistema embarcado, de forma a garantir a consistência de dados entre ambos. As minúcias são pontos formados pelas linhas que estão presentes em todas as impressões digitais. Apesar de não existirem estudos estatísticos suficientes para mostrar a unicidade das minúcias, elas são aceitas como uma forma bastante confiável de comparar impressões digitais.

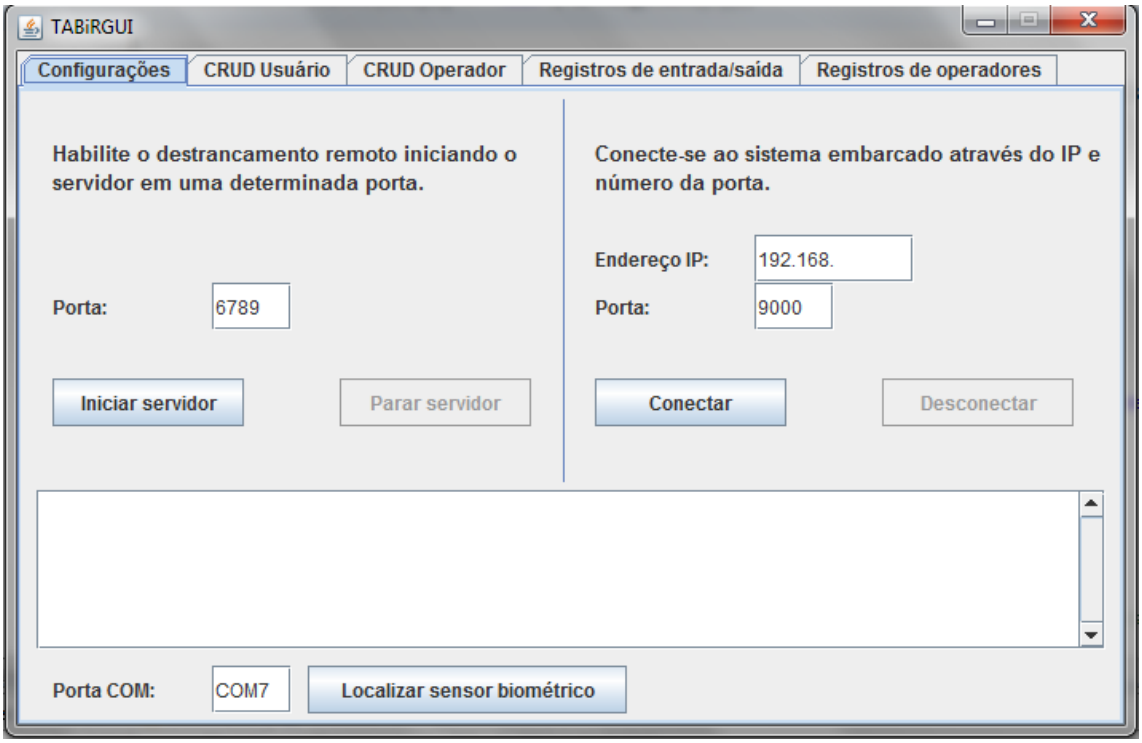

Figura 5 - Aba de configurações do software da estação-base.

A segunda aba consiste de funções de Cadastro, remoção, alteração, busca de usuários (CRUD). Usuários foram definidos como sendo entidades que utilizam o sistema desenvolvido neste projeto com o fim de apenas serem autorizados à entrarem na sala.

Para que o usuário seja cadastrado, os campos Nome, CPF, Cargo, ID devem ser preenchidos, os campos “Login” e “Senha”, utilizados na autenticação via celular não são obrigatórios. Nota-se que o ID a ser inserido deve ser um múltiplo de 4 e menor ou igual a 160, devido às restrições da

memória flash do sensor e do fato de serem cadastradas 4 digitais por usuário (conforme definido previamente nos requisitos do projeto). Além disso, o sensor deve estar conectado à estação-base e já deve ter sido localizado na aba de “Configurações” para que o cadastro de usuários seja habilitado. Após preenchida estas informações, o processo de cadastro das digitais pode ser iniciado utilizando o botão “Inserir”, iniciando com o ID inserido no campo e somando-se 1 a cada digital.

Com isso o usuário utiliza o sensor biométrico instalado na estação-base para o cadastro de suas digitais (Figura 6), seguindo as instruções da tela de cadastro de digitais, e após o término desse cadastro, os dados do usuário são inseridos no banco de dados do projeto, sendo este um arquivo no formato “.txt”, utilizando o método de criptografia DES [4]. As minúcias de cada digital dos usuários são armazenadas em arquivos de extensão “.bin”, sendo um arquivo para cada impressão digital.

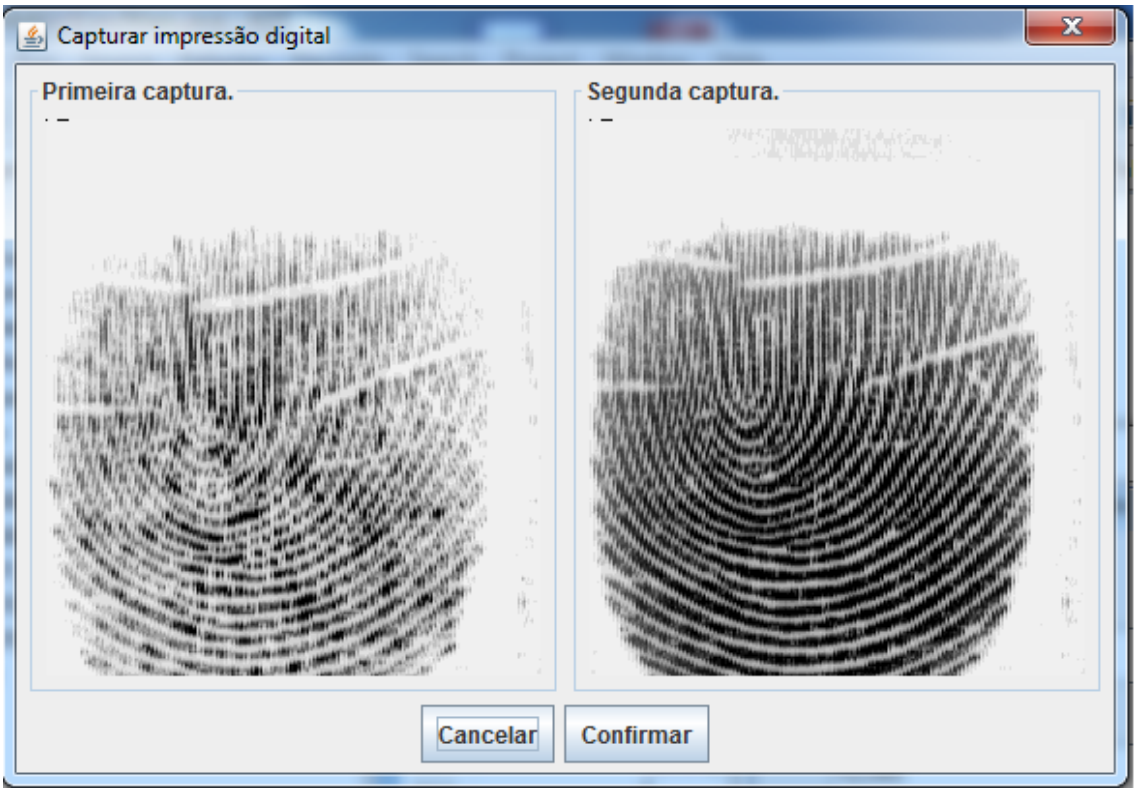

Figura 6 - Tela de captura de impressões digitais.

Também podem ser realizadas as funções de alteração de dados de usuários já existentes no banco de dados, utilizando o botão de “Salvar

alterações”, ou de remoção de usuário do banco de dados, utilizando o botão “Remover usuário”, ou de busca de usuário, através do campo “Nome”. A tela de CRUD de usuários pode ser vista na Figura 7.

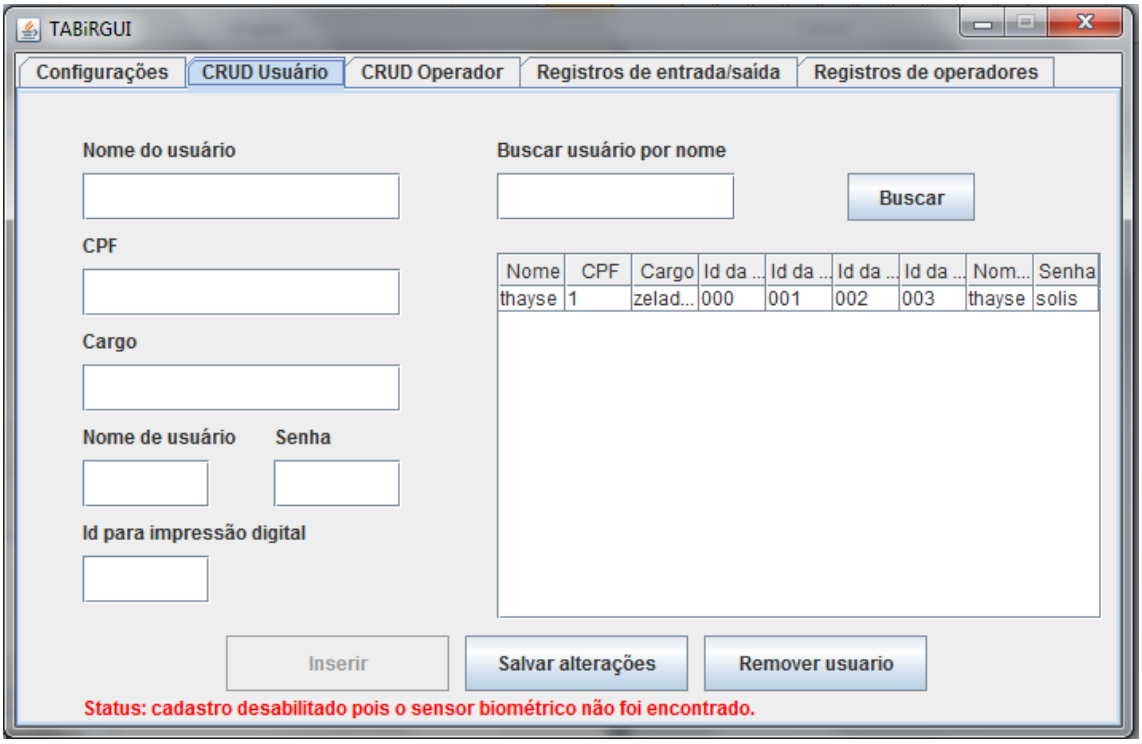

Figura 7 - Aba de “CRUD” de usuários do software da estação-base

A terceira aba consiste das mesmas funções da segunda, entretanto aplicadas aos operadores, portanto sem a necessidade de cadastro de digitais. Para que o cadastro seja realizado, é obrigatório que todos os campos tenham sido preenchidos. Operadores foram definidos como entidades que operam diretamente sobre o software, realizando as funções presentes na primeira aba, além de poder cadastrar usuários ou outros operadores. A interface de terceira aba pode ser visualizada na Figura 8.

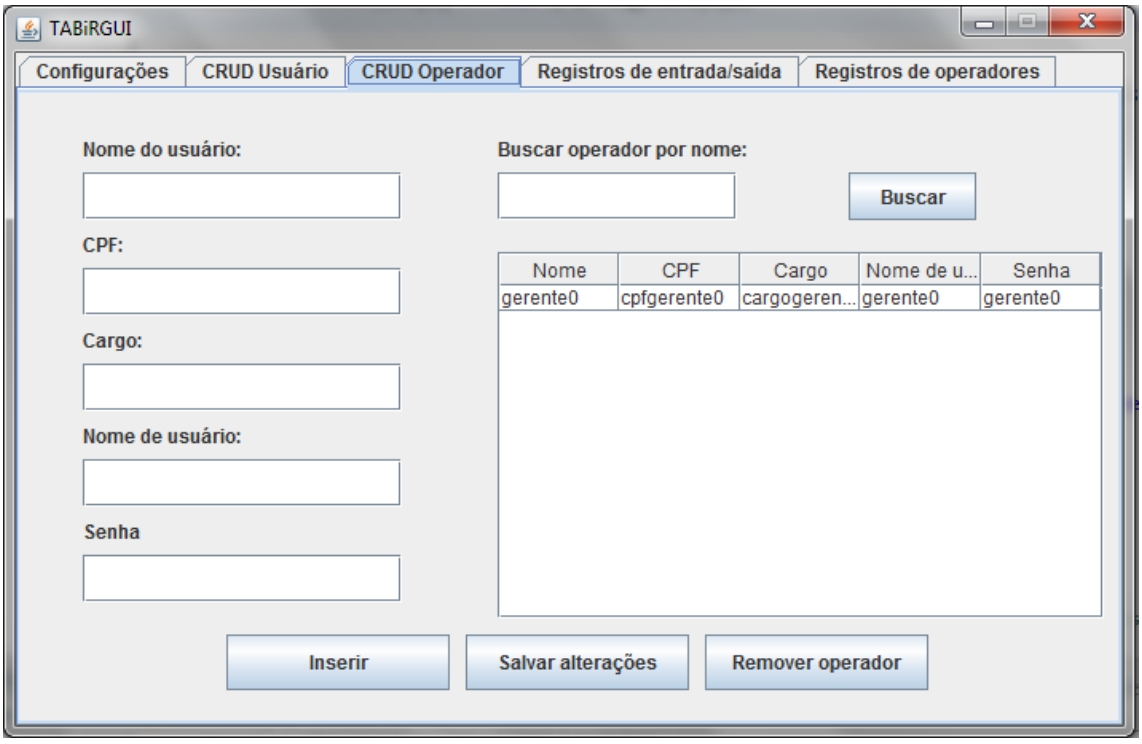

Figura 8 - Aba de “CRUD” para operadores do software da estação-base.

A quarta aba, apresentado na Figura 9, mostra os eventos de entrada/saída, salvos em um arquivo de log em formato texto (.txt). O nome desse arquivo de log é formado pela data (dia, mês, ano e horário) em que o operador entrou no sistema. Assim que os eventos vão surgindo na aba de registros de entrada/saída, o arquivo de log vai sendo atualizado. Dentro desse arquivo de log de eventos de entrada/saída, cada evento é registrado de acordo com o horário em que o pacote, que identifica esse evento, ou a requisição chegou à estação-base.

Para os casos em que houve uma autenticação biométrica no sistema embarcado, tanto o nome de usuário quanto o número ID de sua digital são registrados no log, além do horário. Já para os casos em que houve uma tentativa de autenticação biométrica, é registrada a informação de que houve essa tentativa, além do horário. Em casos de autenticação remota, é registrado o nome de usuário que foi autenticado e o horário de autenticação.

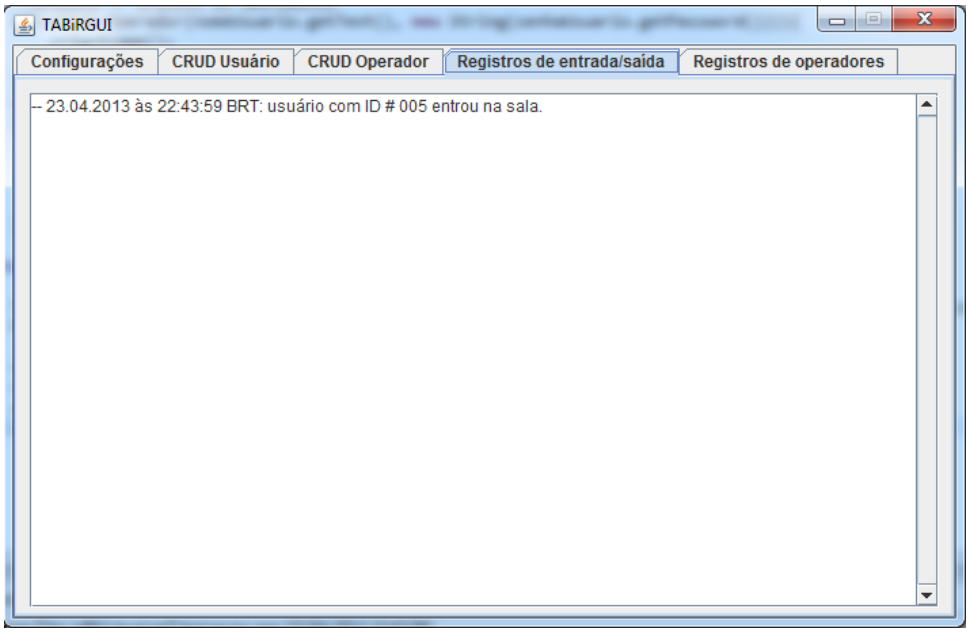

Figura 9 - Aba de registro de entrada e saída do software da estação-base.

A quinta aba, mostrada na Figura 10, consiste em mostrar eventos realizados por operadores. Estes eventos também atualizam o arquivo de log de eventos de operadores. Para cada seção iniciada por um operador é criado um arquivo de log em formato texto (.txt). O nome desse arquivo é formado pelo nome de usuário do operador e a data (dia, mês, ano e horário) em que o operador entrou no sistema.

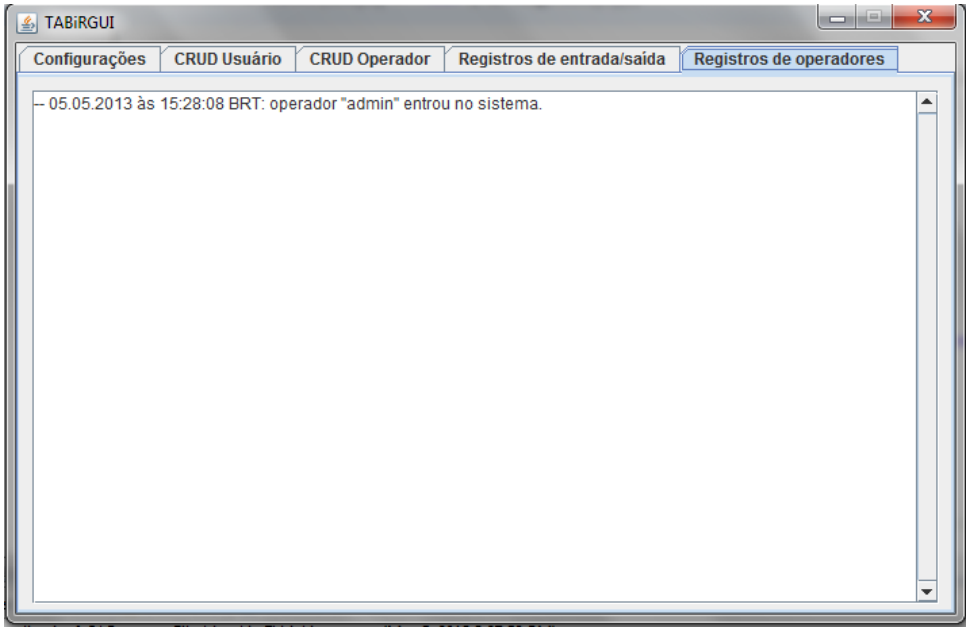

Figura 10 - Aba de registro de eventos de operadores.

5. SISTEMA EMBARCADO

5.1. FIRMWARE

5.1.1. Diagramas

5.1.1.1. Diagrama de casos de uso

Na Figura 11 é mostrado o diagrama de casos de uso do sistema embarcado (que realiza cada um dos casos com o suporte do firmware).

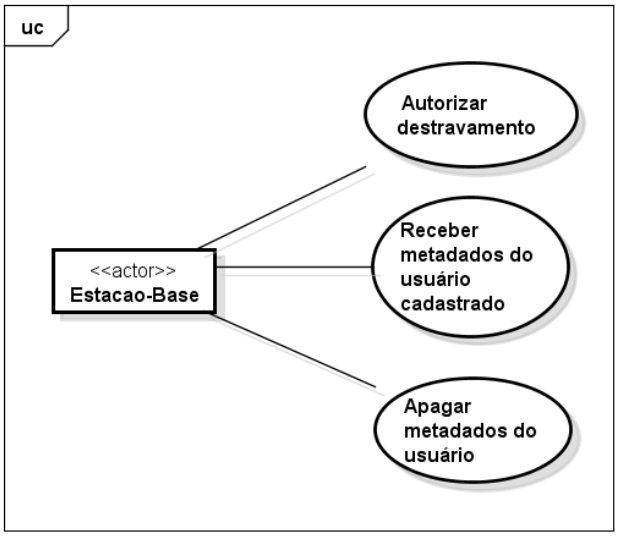

Figura 11 - Diagrama de casos de uso do sistema embarcado

5.1.1.2. Descrição dos casos de uso

2.4. Descrição dos casos de uso do diagrama do sistema embarcado

Caso de uso: Autorizar destravamento

Atores Principais: Estação-base.
Descrição: Executado quando uma tentativa/solicitação de destravamento é detectada.
Pré-condições:

- Tentativa/solicitação de destravamento

Pós-condições:

- Trava destravada

Fluxo Básico:

1. O Sistema reconhece uma tentativa/requisição de destravamento.
2. Se o destravamento é feito

a. Localmente, pelo botão localizado do lado interno da sala. (a)

b. Remotamente, pelo celular, onde a estação-base é que manda a requisição ao sistema embarcado (b)

c. Localmente, pelo sensor biométrico localizado do lado externo da sala (c)

3. O Sistema acende um LED de autorização.
4. O Sistema realiza o destravamento.
5. O Sistema envia informações sobre o destravamento para a estação-base.

Fluxo Alternativo:

a. Pula para o fluxo básico 3.

b. Pula para o fluxo básico 3.

c. O Sistema captura a digital e compara com as existentes na memória do sensor biométrico. Pode haver:

c.1. Falha no reconhecimento da impressão

c.1.1. O Sistema Embarcado acende o LED de negação de autorização

c.1.2. O Sistema Embarcado comunica a tentativa falha de entrada à estação-base.

c.2. Sucesso no reconhecimento da impressão

c.2.1. Pula para o fluxo básico 3.

Caso de uso: Receber metadados do usuário cadastrado

Ator Suporte: Estação-base.

Descrição: Executado quando é feito um cadastro de um novo usuário no banco de dados.

Pré-condições:

- Usuário cadastrado no banco de dados.

Pós-condições:

- Usuário cadastrado no sensor.

Fluxo Básico:

1. O Sistema recebe os metadados do usuário cadastrado da estação-base.
2. O Sistema insere esses na memória do sensor.

Caso de uso: Apagar metadados do usuário removido

Ator Suporte: Estação-base.
Descrição: Executado quando é feito uma remoção de usuário do sistema;
Pré-condições:

- Usuário removido do banco de dados.

Pós-condições:

- Usuário removido do cadastro do sensor.

Fluxo Básico:

1. O Sistema recebe os metadados do usuário cadastrado da estação-base.
2. O Sistema procura os metadados correspondente.
3. O sistema exclui usuário com metadados correspondentes;

5.1.1.3. Diagrama de classes

Na Figura 12 é apresentado o diagrama de classes do firmware. Para uma melhor visualização, recomenda-se que esta figura seja consultada no CD-ROM que acompanha esta monografia.

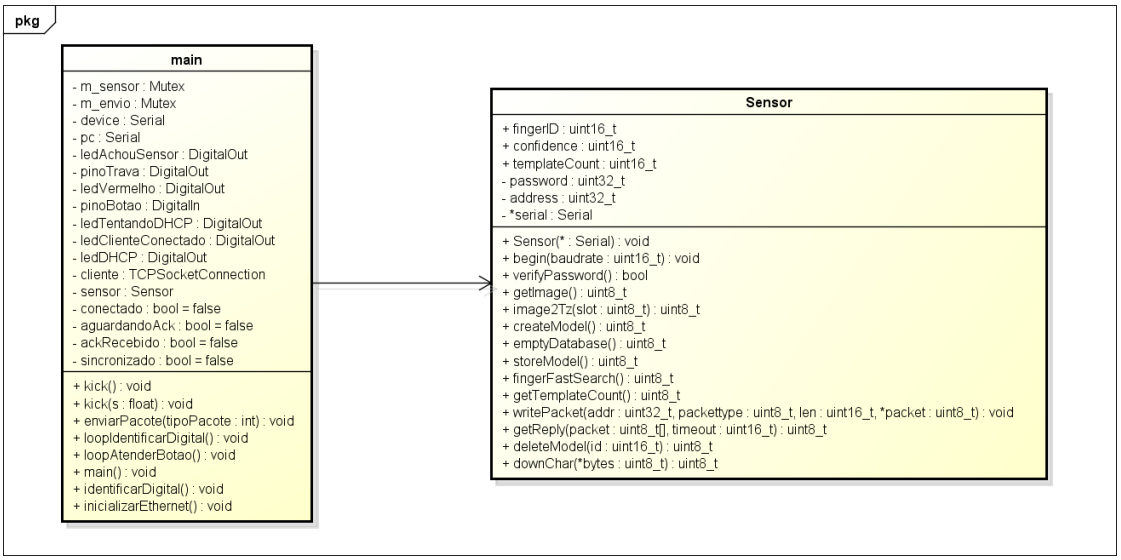
pkg
main
- m_sensor : Mutex
- m_envio : Mutex
- device : Serial
- pc : Serial
- ledAchouSensor : DigitalOut
- pinoTrava : DigitalOut
- ledVermelho : DigitalOut
- pinoBotao : DigitalIn
- ledTentandoDHCP : DigitalOut
- ledClienteConectado : DigitalOut
- ledDHCP : DigitalOut
- cliente : TCPSocketConnection
- sensor : Sensor
- conectado : bool = false
- aguardandoAck : bool = false
- ackRecebido : bool = false
- sincronizado : bool = false
+ kick() : void
+ kick(s : float) : void
+ enviarPacote(tipoPacote : int) : void
+ loopIdentificarDigital() : void
+ loopAtenderBotao() : void
+ main() : void
+ identificarDigital() : void
+ inicializarEthernet() : void
Sensor
+ fingerID : uint16_t
+ confidence : uint16_t
+ templateCount : uint16_t
- password : uint32_t
- address : uint32_t
- *serial : Serial
+ Sensor(* : Serial) : void
+ begin(baudrate : uint16_t) : void
+ verifyPassword() : bool
+ getImage() : uint8_t
+ image2Tz(slot : uint8_t) : uint8_t
+ createModel() : uint8_t
+ emptyDatabase() : uint8_t
+ storeModel() : uint8_t
+ fingerFastSearch() : uint8_t
+ getTemplateCount() : uint8_t
+ writePacket(addr : uint32_t, packettype : uint8_t, len : uint16_t, *packet : uint8_t) : void
+ getReply(packet : uint8_t[], timeout : uint16_t) : uint8_t
+ deleteModel(id : uint16_t) : uint8_t
+ downChar(*bytes : uint8_t) : uint8_t
powered by Astah

Figura 12 - Diagrama de classes do firmware

5.1.1.4. Diagrama de estados

O diagrama de estados do firmware está presente no Apêndice D. Como o diagrama é muito grande, recomenda-se que sua visualização seja feita através da figura original contida no CD-ROM que acompanha esta monografia.

5.1.2. Implementação

O firmware foi implementado em linguagem C++, utilizando a biblioteca de funções do sensor biométrico Adafruit, para Arduino, adaptada para o microcontrolador do projeto. Foi utilizado o ambiente de desenvolvimento online MBED [5], específico para desenvolvimento em microcontroladores ARM, como o utilizado no projeto.

Suas principais funções são: destravar a porta, controlar os LEDs que servem de identificadores visuais de aceitação ou rejeição aos usuários que requisitam o destravamento da porta, além de funções internas como controle de envio e recebimento de pacotes, um *watchdog*, *debounce* para o botão e troca de informações com o sensor, no caso de um cadastro ou remoção de usuário na estação-base, para garantir a sincronização dos dados presentes em ambos os sistemas.

O microprocessador LPC1768 tem um recurso chamado WatchdogTimer. O WatchdogTimer (WDT) é um timer de hardware que reseta o sistema embarcado se o programa principal não resetar seu timer periodicamente [6]. No firmware do sistema embarcado, o Watchdog foi configurado para resetar o sistema embarcado caso ele não tenha sido "servido" em 20 segundos, ou seja, caso seu timer não tenha sido resetado nesse tempo. O reset do watchdog timer vem logo após a captura de uma imagem pelo sensor biométrico. Dessa forma, qualquer problema que possa ocorrer no sensor biométrico, fazendo com que o mesmo fique "travado", não resetará o timer do Watchdog e o sistema embarcado será resetado.

O funcionamento do sistema em alto nível pode ser visualizado nas máquinas de estado do firmware, nas Figuras Figura 13, Figura 14 e Figura 15. Nota-se que foram utilizadas threads para atender as especificações de desempenho esperadas do projeto. Para uma melhor visualização, recomenda-se que estas figuras sejam consultada no CD-ROM que acompanha esta monografia.

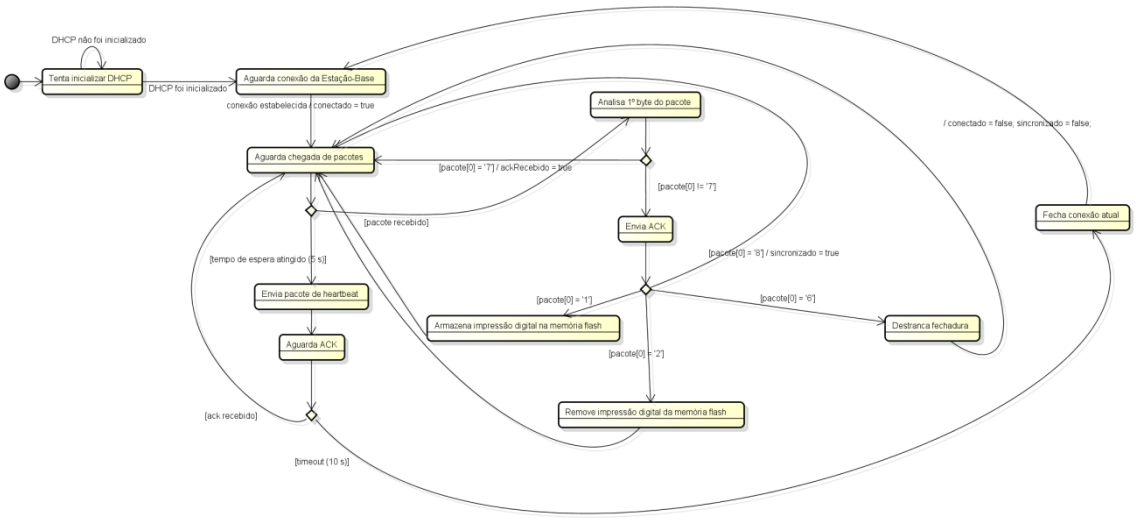
DHCP não foi inicializado
Tenta inicializar DHCP
DHCP foi inicializado
Aguarda conexão da Estação-Base
conexão estabelecida / conectado = true
Aguarda chegada de pacotes
Analisa 1º byte do pacote
[pacote[0] = '7'] / ackRecebido = true
[pacote[0] != '7']
[pacote recebido]
Envia ACK
[pacote[0] = '8'] / sincronizado = true
[tempo de espera atingido (5 s)]
Envia pacote de heartbeat
[pacote[0] = '1']
[pacote[0] = '6']
Armazena impressão digital na memória flash
Destranca fechadura
Aguarda ACK
[pacote[0] = '2']
[ack recebido]
Remove impressão digital da memória flash
[timeout (10 s)]
/ conectado = false, sincronizado = false,
Fecha conexão atual

Figura 13 - Máquina de estados do firmware para recebimento de mensagens

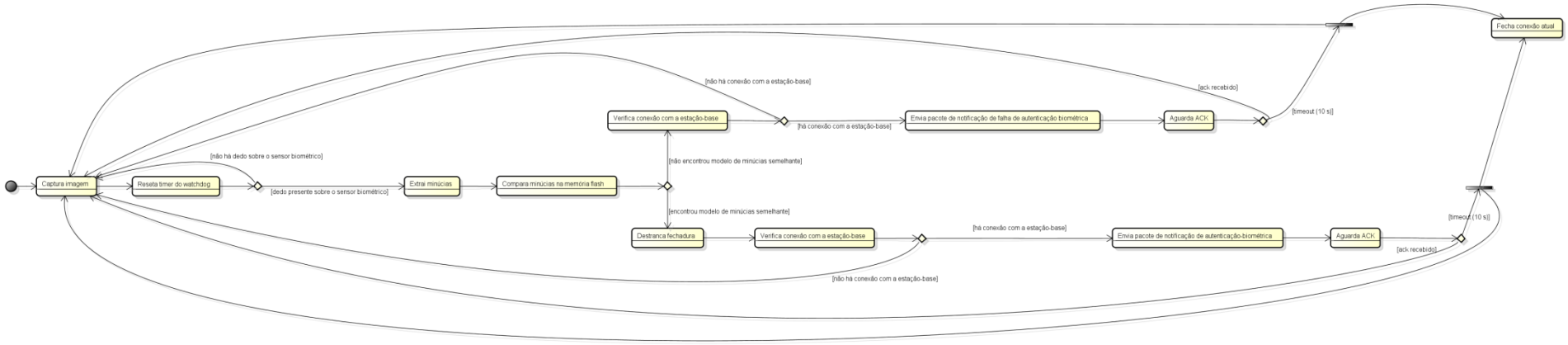
Fecha conexão atual
[não há conexão com a estação-base]
[ack recebido]
Verifica conexão com a estação-base
[há conexão com a estação-base]
Envia pacote de notificação de falha de autenticação biométrica
Aguarda ACK
[timeout (10 s)]
[não há dedo sobre o sensor biométrico]
[não encontrou modelo de minúcias semelhante]
Captura imagem
Reseta timer do watchdog
[dedo presente sobre o sensor biométrico]
Extrai minúcias
Compara minúcias na memória flash
[encontrou modelo de minúcias semelhante]
Destranca fechadura
Verifica conexão com a estação-base
[há conexão com a estação-base]
Envia pacote de notificação de autenticação biométrica
Aguarda ACK
[timeout (10 s)]
[ack recebido]
[não há conexão com a estação-base]

Figura 14 - Máquina de estados do firmware para funcionamento do sensor biométrico.

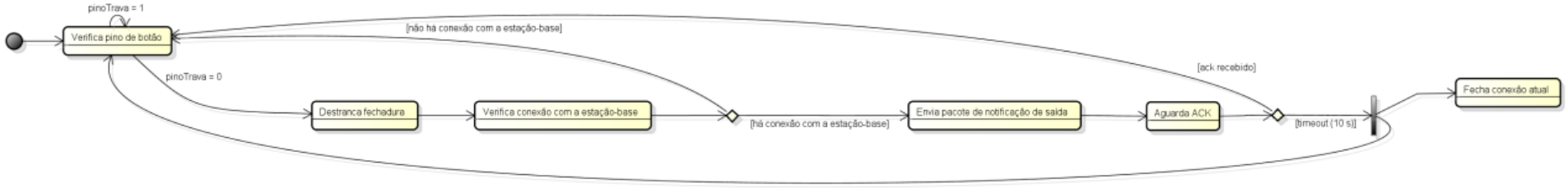

Figura 15 - Máquina de estados do firmware para monitoramento do botão

A máquina de estado apresentada na Figura 13 apresenta o funcionamento do firmware no caso de um recebimento de mensagem.

A máquina de estado da Figura 13 consiste no funcionamento do sensor biométrico no sistema embarcado.

Na Figura 15 está a máquina de estado da *thread* que monitora o botão.

Conforme definido no Plano de Projeto, existem 3 tipos de abertura que podem ocorrer, tratadas de formas diferentes: Biométrica, remota (via celular), e interna (via botão).

O primeiro caso consiste em verificação da presença ou não das minúcias da digital inserida com alguma presente na memória flash, e posterior setagem do pino 21 para zero, em caso de autenticação, que causará via hardware, o destravamento da porta, o acendimento do LED verde e o envio à estação-base de uma mensagem contendo o ID associado a digital analisada. Caso não seja encontrada a digital na memória flash, ou seja, o usuário não está cadastrado, portanto não tem permissão para destravar a porta, o pino 27 é setado para zero, acendendo o LED indicativo vermelho, rejeitando o destravamento e enviando à estação-base uma mensagem constando que houve uma falha na autenticação biométrica.

O segundo caso consiste em a partir de uma mensagem recebida da estação-base (que só será enviada se o usuário remoto que solicitou o destravamento for autenticado pela estação-base), setar o pino 21 para zero, destrancando a porta e enviando a mensagem de “ACK” à estação-base.

O terceiro caso consiste em a partir de uma thread dedicada ao botão, ler o valor do pino conectado ao botão, e se esse valor for zero, setar o valor do pino de destravamento (pino 21) para zero, destravando a porta. Após feito isso o firmware envia uma mensagem à estação-base, notificando-a que houve um destravamento interno.

5.2. HARDWARE

5.2.1. Diagrama em blocos

A Figura 16 mostra o diagrama de blocos com os elementos presentes no hardware.

O sistema embarcado então foi dividido em três subgrupos: circuito de alimentação, circuito da trava e circuito do microcontrolador, sendo que o circuito do microcontrolador e o circuito da trava, possuem alimentações separadas e independentes, mas idênticas no funcionamento. Os três subgrupos serão discutidos no próximos tópicos.

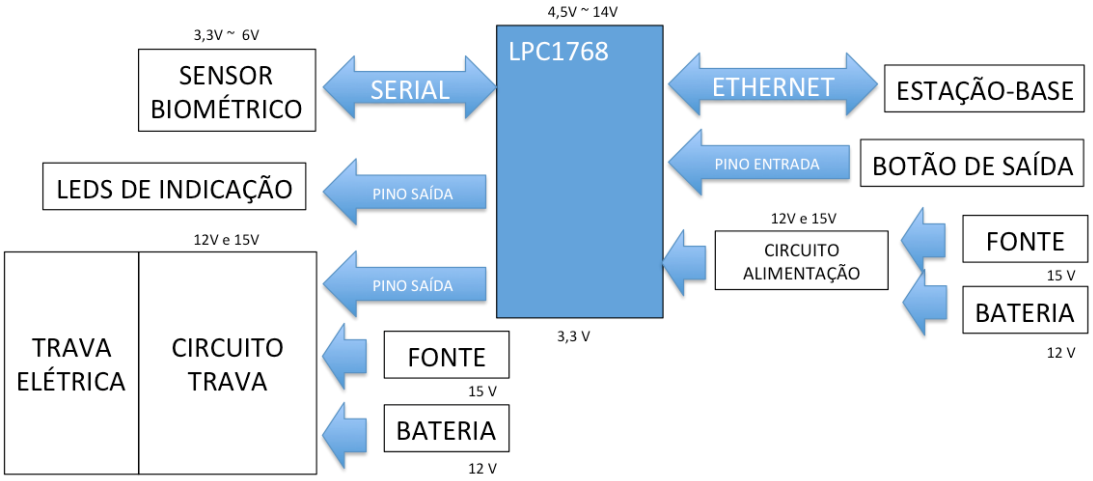

Figura 16 - Diagrama de blocos do hardware

5.2.2. Circuito do microcontrolador

Conforme discutido na análise de tecnologias, o microcontrolador empregado é o LPC1768. Ele requer 5V de alimentação. Dessa forma, são empregados dois reguladores LM7805, ligados em paralelo, que regulam os 12 V provenientes da bateria do circuito de alimentação e fornecem 5V, assim com a corrente necessária para o funcionamento de todos os elementos do circuito.

O circuito em que se encontra o microcontrolador foi isolado do circuito em que está a trava. Para o acionamento dela é necessária uma corrente alta (4 A) por se tratar de um componente eletromagnético, e o processador trabalha com correntes mais baixas (menores que 500 mA). Para esse fim, utilizou-se um optoacoplador cujo esquema é mostrado na Figura 17.

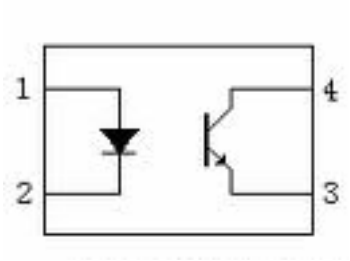

Figura 17 - Esquema do optoacoplador

O funcionamento desse dispositivo é simples: quando um LED (presente em seu interior) é polarizado e emite luz, um fototransístor (contido também no encapsulamento do optoacoplador) tem sua pastilha exposta ao estímulo luminoso e passa a conduzir, operando como uma chave. Esse tipo de circuito é muito utilizado quando se deseja um isolamento elétrico entre duas partes de um circuito.

No que diz respeito a conexão física entre os módulos para comunicação, foi usado um conector fêmea RJ45 com transformador ligado ao microcontrolador nos pinos Ethernet RD+ ,RD-, TD+ e TD-. O conector RJ45, por sua vez, foi conectado a um cabo ligado a um roteador presente na rede local.

Para a conexão com o sensor de reconhecimento biométrico foram utilizados os pinos serial Rx e Tx do microprocessador. O sensor precisa de 5V para seu funcionamento, sendo ele um dos elementos ligados a saída do regulador LM7805.

O botão, presente no circuito do sistema embarcado, é utilizado para liberar a trava do lado de dentro da sala. Para cumprir tal tarefa foi acrescentado um push button ao circuito. Enquanto ele não for pressionado, o pino do microcontrolador responsável pelo seu monitoramento receberá nível lógico alto e quando pressionado, o pino passará para o nível lógico baixo. O pressionamento do push button informa ao firmware que deve ser executada a rotina de abertura da trava (em que um outro pino – de saída – tem seu nível lógico alterado, estimulando o optoacoplador que liga os dois circuitos).

Por fim há também os leds indicadores de autenticação bem-sucedida (verde) ou mal-sucedida (vermelho). Um de seus terminais estão ligados aos 3,3 V de saída fornecido pelo microcontrolador e mudam de estado (acendem)

assim que o outro terminal a que estão ligados passa para o nível lógico baixo, permitindo a passagem de corrente.

5.2.2.1. Sensor biométrico

O sensor utilizado recebe comandos e transfere dados por serial. Por padrão BAUD_RATE = 57600, 8 databits, 1 startbit, 1 stopbit, sem paridade. A comunicação com o sensor começa com a mensagem VFYPWD na qual manda-se uma senha para ele. A senha padrão é 0x00000000.

O sensor possui um buffer de imagem onde cabe uma digital de 256x288. Além disso, há 2 buffers de chars (Bytes) cada um com 512 B. O procedimento de captura de uma digital envolve a coleta da imagem do digital e o armazenamento no buffer de imagem. O conteúdo dos buffers é volátil.

Existe uma operação que extrai informações sobre a imagem no buffer de imagem e coloca em um dos buffers de char. O "modelo" da digital é gerado combinando as informações extraídas de 2 coletas diferentes da mesma digital. O sensor não armazena as imagens das digitais na sua flash, apenas os modelos.

Para gravar uma digital na base de dados do sensor é preciso primeiramente haver um modelo. Para que este seja gravado, ele deve estar num dos char buffers. Para que isso ocorra, ou ele foi gerado anteriormente a partir de uma digital ou foi recebido de algum lugar (como é o caso do sensor presente no sistema embarcado).

O procedimento de geração do modelo tem as seguintes etapas:

1) Coleta da imagem da digital (no buffer de imagem)

2) Extração das informações no char buffer 1.

3) Coleta da mesma digital novamente.

4) Extração doas informações no char buffer 2.

5) Combinação das informações dos char buffers 1 e 2 em um modelo unificado, que ocupa os char buffers.

O software da Estação-Base realiza os cinco passos anteriores, mas não salva o modelo na flash do seu sensor pois não é necessário. Ele também transfere do sensor o modelo (conteúdo dos 2 buffers de chars).

O software embarcado deve receber da estação base via socket o modelo (não as imagens). Em seguida, este é enviado aos char buffers do sensor do sistema embarcado. Após, o sistema embarcado manda o sensor gravar o modelo em sua flash.

Para identificar uma digital, captura-se sua imagem via sensor, são extraídas as informações em um dos char buffers e procura-se na flash um modelo que corresponda a digital. O sensor vai responder ou que não achou o modelo ou o número do modelo que ele achou e o grau de confiabilidade.

5.2.3. Circuito da trava

Este circuito tem dois componentes de destaque: o optoacoplador e o relé. O optoacoplador tem a função de isolar a corrente que passa do microcontrolador para a trava.

O relé é utilizado para acionar a trava a partir da bateria. A bobina do relé é saturada no momento que o transistor do optoacoplador está saturado; assim, ele é referenciado ao terra, gerando um fluxo de corrente. Enquanto o pino do microcontrolador tem sua saída em nível lógico alto, não há passagem de corrente no optoacoplador, pois não existe diferença de potencial entre os dois pontos de entrada do circuito da trava. Quando o microcontrolador seta o pino de acionamento para nível lógico baixo, a trava é acionada.

5.2.3.1. Trava elétrica

A fechadura elétrica (Figura 18) é um dispositivo normalmente utilizado na entrada de edifícios onde se requer um nível mínimo de segurança.

Figura 18 - Trava elétrica

Seu funcionamento é baseado em uma parte mecânica e uma magnética, conforme pode ser observado na Figura 19.

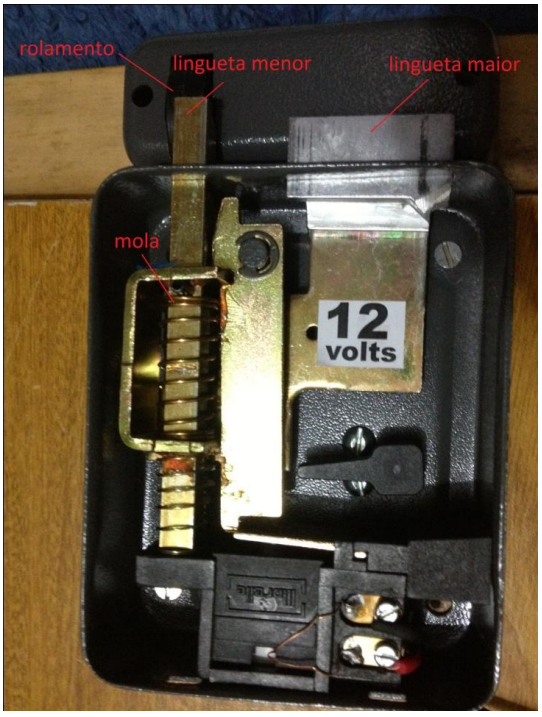

Figura 19 - Interior da trava elétrica

Quando há uma corrente elétrica passando pelo indutor presente no interior da fechadura, este gera um campo magnético, que ativa a trava interna e dispara um processo mecânico da porta.

Na parte mecânica há uma mola que fica comprimida enquanto a porta estiver fechada em contato com o batente (dependendo da lingueta menor). A lingueta maior, depende da mola, e em consequência é também dependente da lingueta menor. Enquanto a mola não estiver comprimida, a lingueta maior

tem liberdade total para se movimentar para fora ou para dentro. A partir do momento em que a mola é comprimida, a lingueta maior permanece fora da fechadura sem mobilidade.

Enquanto a porta estiver fechada e não houver uma corrente elétrica passando pela bobina, a mola não tem como sair do estado de compressão, mantendo a lingueta maior para fora e a porta fechada.

Se a porta estiver aberta e a bobina não-saturada, a lingueta menor tem mobilidade, a mola fica expandida; a lingueta maior também tem mobilidade. Para esse caso, se a trava for liberada, nada acontece pois todos os elementos internos já estão liberados.

Se a porta estiver fechada e a bobina for acionada (com uma passagem de corrente), a mola é expandida e a lingueta maior é liberada. A função da lingueta menor, para esse caso é deslocar a porta no sentido da sua abertura. No momento em que a lingueta maior é liberada, não existe nada que impeça a porta de abrir. Quando a porta está fechada, a lingueta menor é pressionada por um rolamento que gera uma forca centrípeta que empurra a lingueta para fora do rolamento.

5.2.4. Circuito de alimentação

De acordo com a especificação, o sistema embarcado deve manter seu funcionamento por algum tempo no caso de falta de energia na rede elétrica, acionando uma bateria automaticamente ao ser detectado tal comportamento.

A estratégia utilizada foi um circuito conhecido como circuito caixa d'água. Todo o sistema sempre é alimentado pela bateria (independente do funcionamento da rede elétrica), porém juntamente com a alimentação fornecida pela bateria, existe um circuito de monitoramento alimentado por uma fonte, que verifica se a bateria está carregada. Caso ela não esteja, o próprio circuito de monitoramento carrega a bateria, fazendo com que ela se mantenha carregada enquanto a rede elétrica estiver ligada.

O circuito de monitoramento tem como elemento principal o LM555. O LM555 consiste em dois comparadores de tensão, um flip-flop, um transístor de descarga e uma malha divisora resistiva usada para fixar os níveis de tensão dos comparadores (Figura 20).

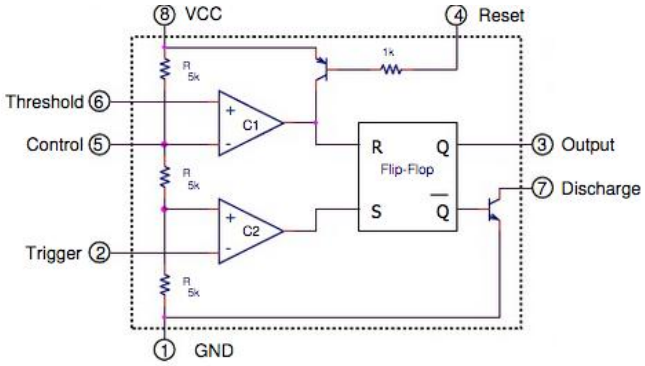

Figura 20 - Esquema interno do LM555.

Como as três resistências são de valor igual (5KΩ), o comparador superior (C1) é internamente referenciado a 2/3 da tensão VCC e o Comparador C2 é referenciado a 1/3 de VCC. As saídas dos dois comparadores estão ligadas às entradas do flip- flop RS, que definem se a saída (pino 3) está no estado alto ou baixo. Existe ainda uma saída complementar (pino 7), que está ativa quando pino 3 está no nível baixo e vice-versa. Os níveis de comparação de 1/3 e 2/3 de VCC existem quando o pino 5 (Control) não se encontra ligado.

Neste projeto foi utilizada uma tensão de referência de 12V no pino 5 fornecida pelo diodo zener; logo é necessário fornecer uma tensão de entrada aos pinos 2 e 8 que indicam ao LM555 a tensão de corte e acionamento do flip-flop. Se a bateria começar a descarregar, as tensões diminuirão, implicando que os comparadores setem o flip-flop, liberando tensão para o carregamento da bateria. A saída do LM555 foi ligada a um MOS-FET que ao ser acionado libera a tensão de 15V para carregar a bateria.

Tanto o circuito de alimentação do microprocessador quanto o da trava funcionam da mesma maneira.

5.2.5. Diagramas elétricos

A Figura 21 mostra o diagrama elétrico do microcontrolador.

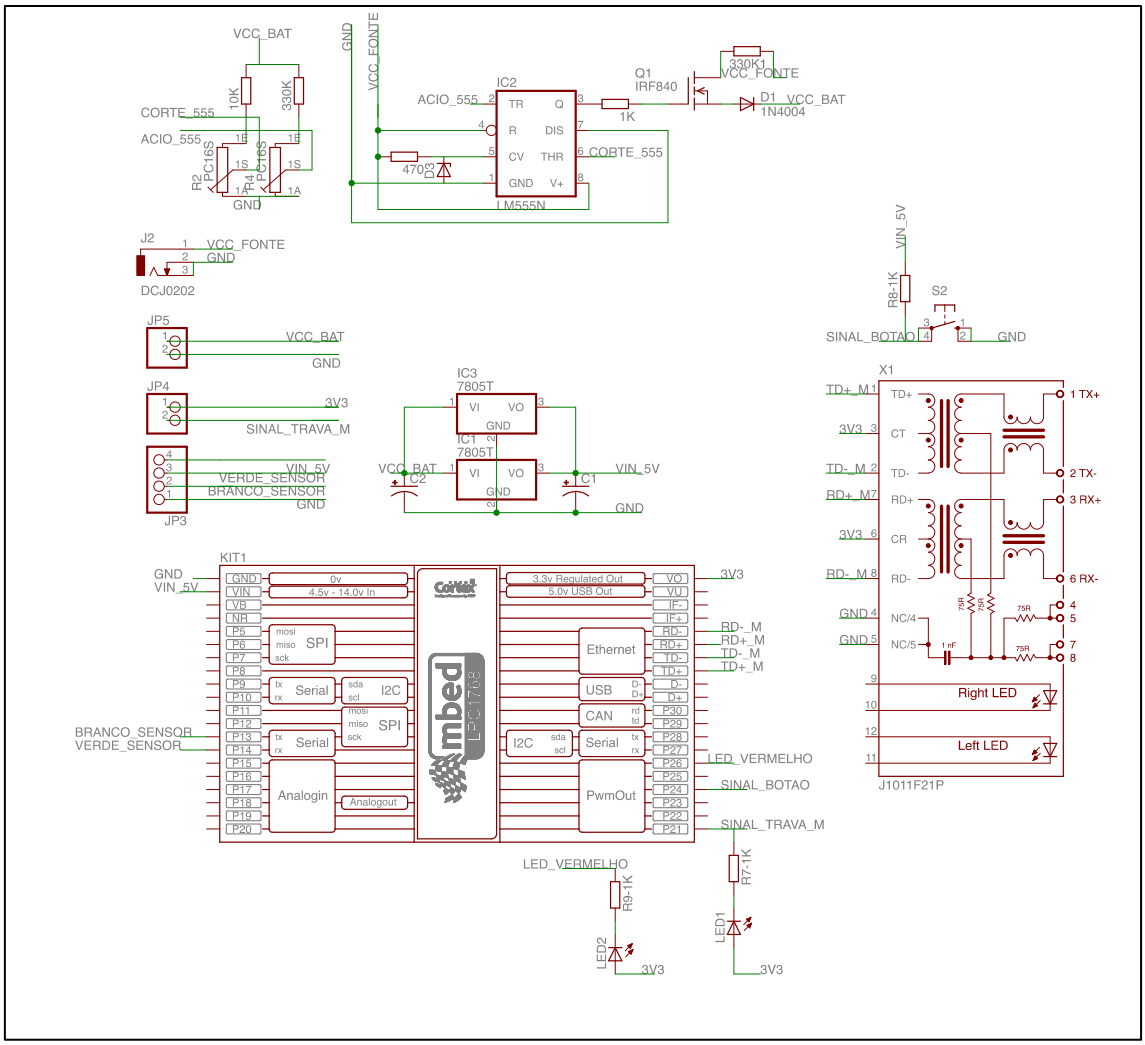

Figura 21 - Esquemático do circuito do microcontrolador

A Figura 22 mostra o diagrama elétrico do circuito da trava.

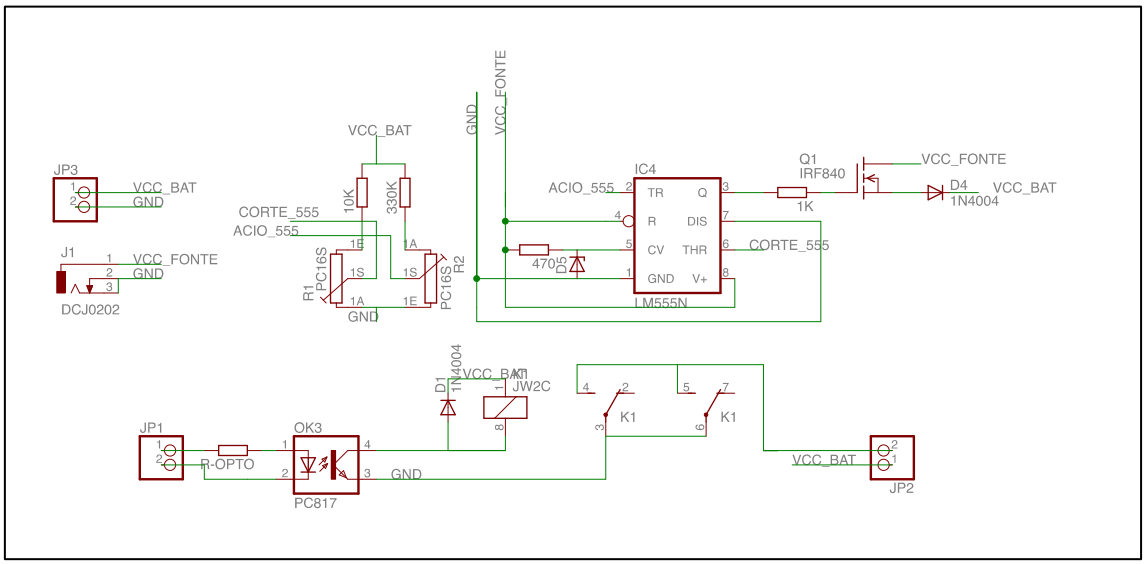

Figura 22 - Esquemático do circuito da trava.

5.2.6. Projeto das PCBs

Foram elaboradas duas placas de circuito impresso no intuito de minimizar interferências entre componentes. A partir dos diagramas elétricos foram elaboradas as PCBs equivalentes. A Figura 23 mostra a PCB do circuito do microcontrolador. Em seguida, são mostradas a parte superior (Figura 24) e inferior (Figura 25) da PCB do circuito da trava, que é dupla-face. Os desenhos são representativos e não indicam o tamanho real da placa.

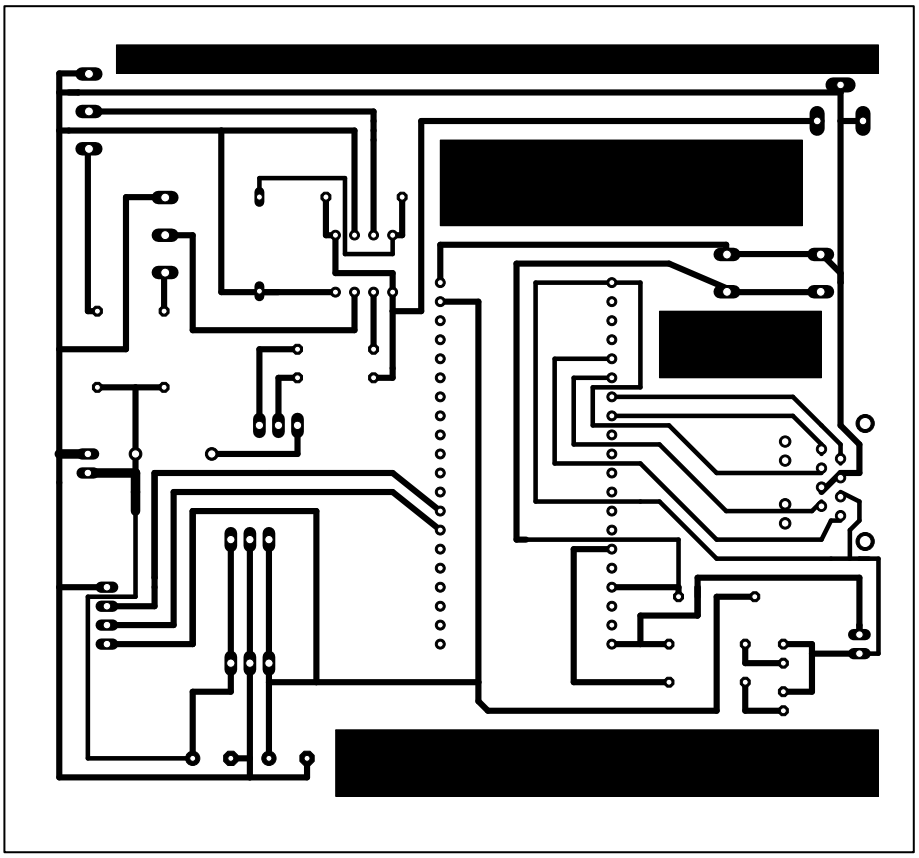

Figura 23 - PCB do circuito do microcontrolador

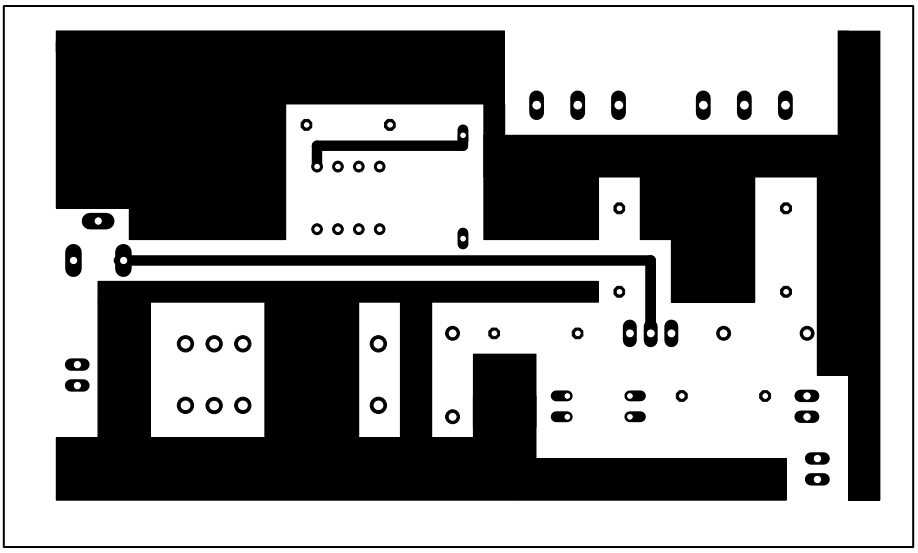

Figura 24 - Parte superior da PCB do circuito da trava.

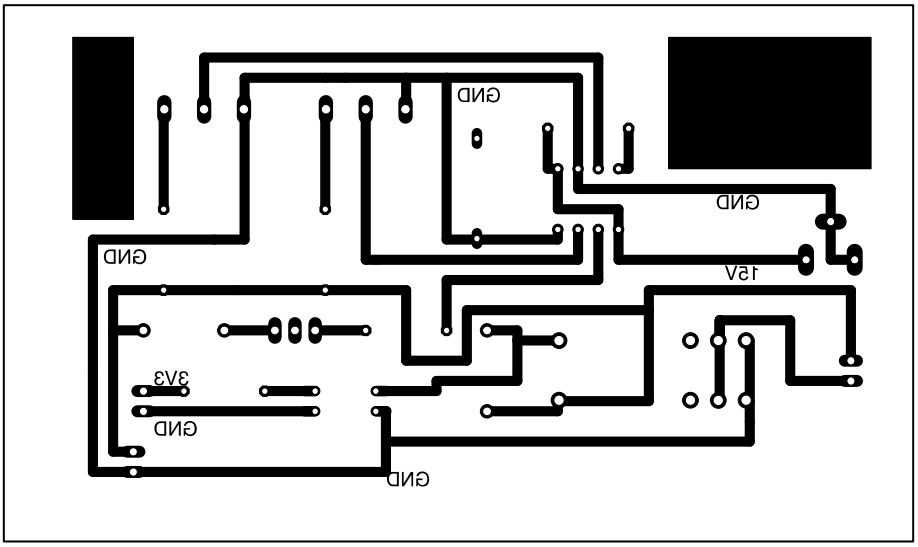

Figura 25 - Parte inferior da PCB do circuito da trava

5.2.7. Lista de componentes para confecção da PCB completa

Na Tabela 10 são listados os componentes necessários para a confecção do circuito do microcontrolador.

Tabela 10 - Lista de componentes para confecção do circuito do microcontrolador

Qtde	Componentes
1	LPC1768
1	Diodo Zener 12V

2	Diodo 1N4004
2	Potenciômetros 500KΩ
1	LM555
4	1KΩ
1	10KΩ
1	330KΩ
1	470KΩ
2	LEDs
1	MOS-FET (IRF840)
2	Regulador de Tensão (LM7805)
2	Dissipadores
2	Capacitor Eletrolítico 10μF
1	Push Bottom
1	Conector RJ45
1	Jack (Conector bateria)
1	Borne (Conector fonte)
2	Pinos Fêmea
4	Pinos Macho

Na Tabela 11 são listados os componentes necessários para a confecção do circuito da trava.

Tabela 11 - Lista de componentes para confecção do circuito da trava

Qtde	**Componentes**
1	Relé 12V 1.5A (JW2C)
1	Diodo Zener 12V
2	Diodo 1N4004
2	Potenciômetros 500KΩ
1	Temporizador (LM555)
1	1KΩ
1	10KΩ

1	330KΩ
1	470KΩ
1	180KΩ
1	MOS-FET (IRF840)
1	Optoacoplador (PC817)
1	Jack (Conector bateria)
1	Borne (Conector fonte)
2	Pinos Fêmea

5.2.8. Guia de montagem

A montagem do sistema embarcado depende dos componentes listados na Tabela 10 e Tabela 11 além dos seguintes componentes:

- 2 Fontes 15V;
- 1 Bateria 12V 1,5A/h;
- 1 Bateria 12V 7A/h;
- 2 cabos para as baterias;
- 1 cabo fêmea-fêmea para dois pinos;
- 1 sensor biométrico;
- 1 trava elétrica;
- 1 cabo de par trancado direto

Passo 1 – Regular tensões de corte e acionamento do LM555

Para cada uma das placas conectam-se a fonte e a bateria e verifica-se os valores de tensão presentes no pino 6 e no pino 2, pinos de corte e de acionamento respectivamente. O pino 6 deve apresentar a tensão de aproximadamente 12,6V e o pino 2, aproximadamente 6V.

Caso estejam desregulados, deve-se se ajustar os potenciômetros (Figura 21) para cada um dos pinos.

Passo 2 – Conexão dos elementos

Após realizar o passo 1, os circuitos devem ser desligados da bateria e fonte, para serem conectados os demais elementos.

- Sensor: para conectar o sensor, deve-se observar na placa qual é o pino de 5V e qual se refere ao terra; o cabo vermelho do sensor deve ser ligado ao pino 5V e o preto com o terra.
- Microcontrolador: deve ser instalado de forma que o conector USB fique ao lado dos potenciômetros.
- Cabo fêmea-fêmea: deve ser instalado ligando a placa do microcontrolador com a placa da trava. Deve-se prestar atenção pois não é uma ligação direta, o cabo deve ser cruzado para que ligue o 3,3V da placa do microprocessador com os 3,3V da placa da trava.
- Trava elétrica: deve ser conectada no circuito da trava no borne localizado ao lado do relé.
- Cabo Ethernet: é conectado no conector RJ45 na placa do microcontrolador.

Passo 3 – Conexão da alimentação

Deve-se ligar primeiramente a bateria e em seguida a fonte.

Baterias: para ligar a bateria basta observar na placa qual é o lado onde o conector se conecta com o terra e qual o lado que se conecta a 12V. O pino preto da bateria deve ser ligado ao terra, e o vermelho aos 12V. O borne encaixa-se facilmente ao lado correto das respectivas tensões. Para o circuito da trava, deve ser ligada a bateria de 7A/h (bateria maior) e para o circuito do microcontrolador deve ser ligada a bateria de 1,3A/h.

Fontes: devem ser conectadas diretamente aos pinos jack das duas placas.

6. CELULAR

6.1. DIAGRAMAS

6.1.1. Diagrama de classes

Na Figura 26 é apresentado o diagrama de classes do celular.

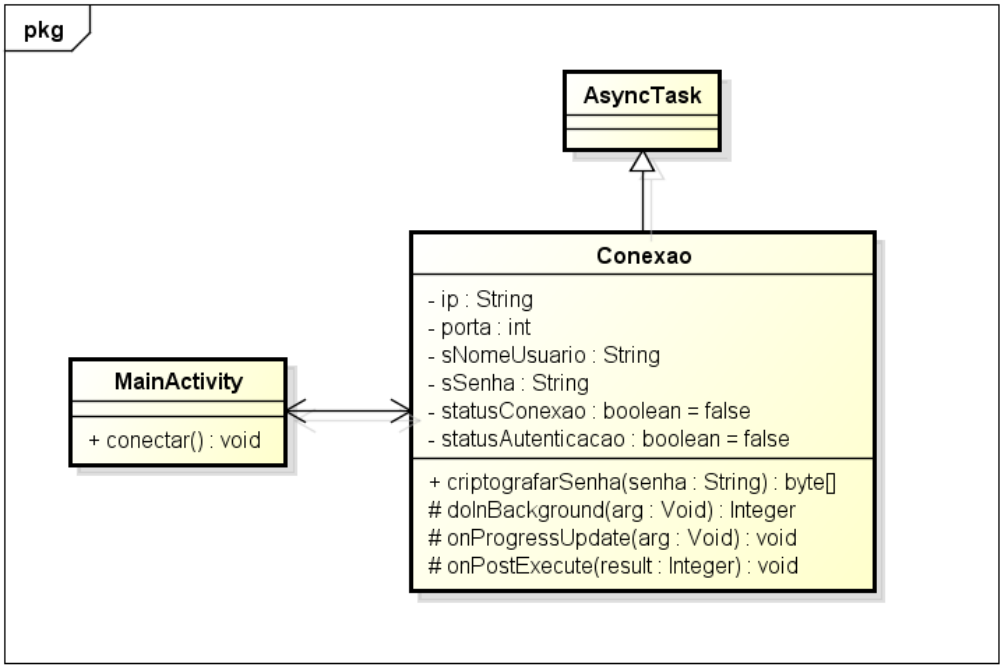
pkg
AsyncTask
Conexao
- ip : String
- porta : int
- sNomeUsuario : String
- sSenha : String
- statusConexao : boolean = false
- statusAutenticacao : boolean = false
+ criptografarSenha(senha : String) : byte[]
# doInBackground(arg : Void) : Integer
# onProgressUpdate(arg : Void) : void
# onPostExecute(result : Integer) : void
MainActivity
+ conectar() : void

Figura 26 - Diagrama de classes do celular

6.1.2. Diagramas de casos de uso

Na Figura 27 é apresentado o diagrama de casos de uso do celular.

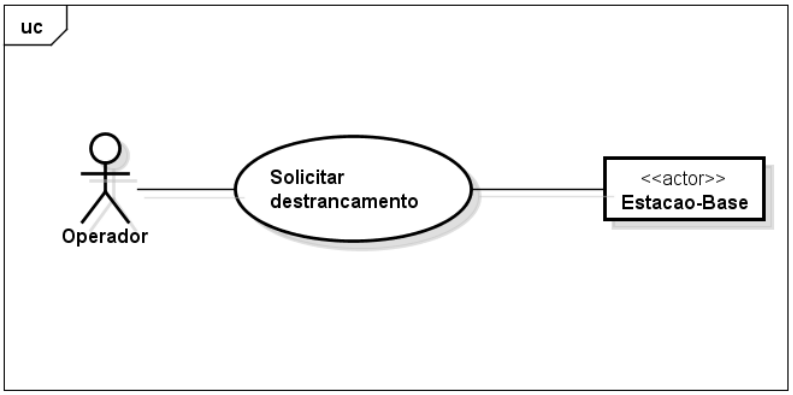

Figura 27 - Diagrama de caso de uso do celular

6.1.3. Descrição do caso de uso

6.2. *Caso de uso: Solicitar destrancamento*

Ator Principal: Usuário

Descrição: Executado quando deseja-se executar um destrancamento remotamente.

Pós-condições:

- Recebimento de mensagem de autorização ou rejeição.

Fluxo Básico:

1. O usuário preenche as informações pedidas pelo software de acordo com as regras de negócio.
2. O usuário confirma as informações.
3. O Sistema manda uma requisição para a estação-base.
4. O Sistema recebe uma mensagem de autorização ou rejeição.

Regras de negócio:

1. Todos os campos são obrigatórios.

6.3. INTERFACE E USO DO SOFTWARE

O software móvel foi desenvolvido para a plataforma Android por razões expostas no plano de projeto. O software oferece suporte para versões no mínimo 2.2 até 4.1 do sistema operacional Android. Não se garante a compatibilidade do software com versões Android acima da 4.1. Essas informações foram retiradas da própria ferramenta de desenvolvimento que está sendo utilizada pela equipe.

O software do celular permite o destravamento da porta remotamente, caso a estação-base autentique o usuário. Nesse caso, será mostrada uma mensagem no aplicativo indicando se o destravamento foi ou não foi realizado, por algum motivo.

Caso a estação-base rejeite o destravamento devido a um usuário não cadastrado, é mostrada uma mensagem indicando que não foi possível destravamento devido a problemas de autenticação.

Problemas de conexão com o servidor ou campos em branco são indicados através de mensagens de erro também.

A interface do aplicativo móvel, que pode ser observada na Figura 28, é constituída de quatro campos onde é possível inserir o endereço IP do servidor (estação-base), porta para conexão, nome de usuário e senha, um botão “Destrancar porta” e um campo de texto (“Status”) que mostra mensagens indicando erros (conexão com o servidor, sem acesso à Internet, entre outros), se o usuário foi autenticado ou não e se a fechadura foi destrancada ou não.

Figura 28 - Interface do aplicativo móvel

7. PROTOCOLO DE COMUNICAÇÃO

7.1. COMUNICAÇÃO ENTRE ESTAÇÃO-BASE E CELULAR

A comunicação entre estação-base e celular foi implementada utilizando um modelo cliente-servidor. Isso foi possível através do uso de sockets sobre o protocolo TCP, em que a estação-base é o servidor e o celular é o cliente. O software da estação-base, que representa o servidor, possui opções para iniciar o servidor, onde é aceita uma conexão remota, e parar o servidor, onde a comunicação remota é desativada.

Na aba de configuração do servidor, é possível escolher em qual porta o servidor ficará escutando conexões. Nota-se que no roteador para o qual o servidor está conectado, esta porta deverá estar aberta (*port forwarding*), caso contrário o próprio roteador barrará conexões externas (por questões de segurança). Assim que o servidor for iniciado, o mesmo ficará escutando por conexões até que o operador feche o software ou pare o servidor, ou algum motivo externo interrompa o funcionamento do servidor, por exemplo, uma queda de conexão com a Internet ou um desligamento do roteador ao qual o servidor está ligado.

O servidor aceita uma conexão remota por vez, ou seja, a implementação de sockets não é multi-*thread*. Através do endereço IP do servidor e a porta pela qual o servidor está escutando por conexões, é possível utilizar o aplicativo móvel para enviar um nome de usuário e senha e requisitar o destrancamento da trava.

Assim que as informações de autenticação são preenchidas no celular, clica-se no botão “Destrancar porta” para se conectar ao servidor e enviar usuário e senha (criptografada) inseridos no aplicativo móvel, e aguarda-se, num tempo limite de quinze segundos, que o servidor autentique a requisição. Caso não seja possível uma conexão com o servidor ou o tempo limite expire, uma mensagem de erro é mostrada no aplicativo móvel. O usuário é enviado como um array de bytes e o servidor converte a informação para uma cadeia de caracteres (String); a senha é criptografada antes de ser enviada ao servidor. Ambos são validados através de uma busca no banco de dados do servidor.

A requisição de destrancamento está implícita, ou seja, assim que o servidor receber a requisição de conexão remota, o mesmo iniciará o processo de validação do nome de usuário e senha enviados, a fim de destrancar a fechadura. Se o usuário e senha forem inválidos, a estação-base envia uma mensagem de não autenticação para o celular.

Caso o usuário e senha sejam válidos, três situações distintas podem ocorrer: há conexão com o sistema embarcado e o pacote de destrancamento da fechadura foi enviado com sucesso; há conexão com o sistema embarcado, mas o envio do pacote de destrancamento falhou e, por último, não há conexão com o sistema embarcado.

Cada uma das situações resulta em uma mensagem distinta que será enviada ao celular. O celular recebe cada mensagem e, através de uma variável do tipo inteiro (denominada resultado), mostra a mensagem adequada ao usuário.

A Tabela 12 relaciona quais mensagens são enviadas da estação-base para o celular de acordo com a situação, quais são as mensagens resultantes que serão mostradas ao usuário do celular e o valor da variável que controla essas mensagens.

Tabela 12 - Mensagens enviadas da estação-base para o celular e descrição de cada uma delas ao usuário final

Situação	**Mensagem enviada da estação-base para o celular**	**Valor da variável "resultado"**	**Mensagem mostrada no celular**
Usuário válido, há conexão com o sistema embarcado e o pacote de destrancamento foi enviado com sucesso.	"OK"	0	"Usuário foi autenticado e a fechadura foi destrancada."
Usuário válido, há conexão com o sistema embarcado e falhou o envio do pacote de destrancamento.	"NOK"	3	"Envio do pacote ao sistema embarcado falhou."

Usuário válido, mas não há conexão com o sistema embarcado.	“SEDESCONECTADO”	4	“Não há conexão entre estação-base e sistema embarcado.”
Usuário inválido.	“USUARIOINVALIDO”	1	“Erro ao se autenticar.”

Caso estoure o tempo limite de aguardo no aplicativo móvel (de quinze segundos), a variável resultado assume valor igual a 2 e a seguinte mensagem é mostrada ao usuário: “Tempo limite esgotado”.

A Figura 29 mostra a máquina de estados para a requisição de destrancamento da fechadura, enviada pelo celular. Já a Figura 30 mostra a máquina de estados da estação-base, na situação em que uma requisição remota será atendida.

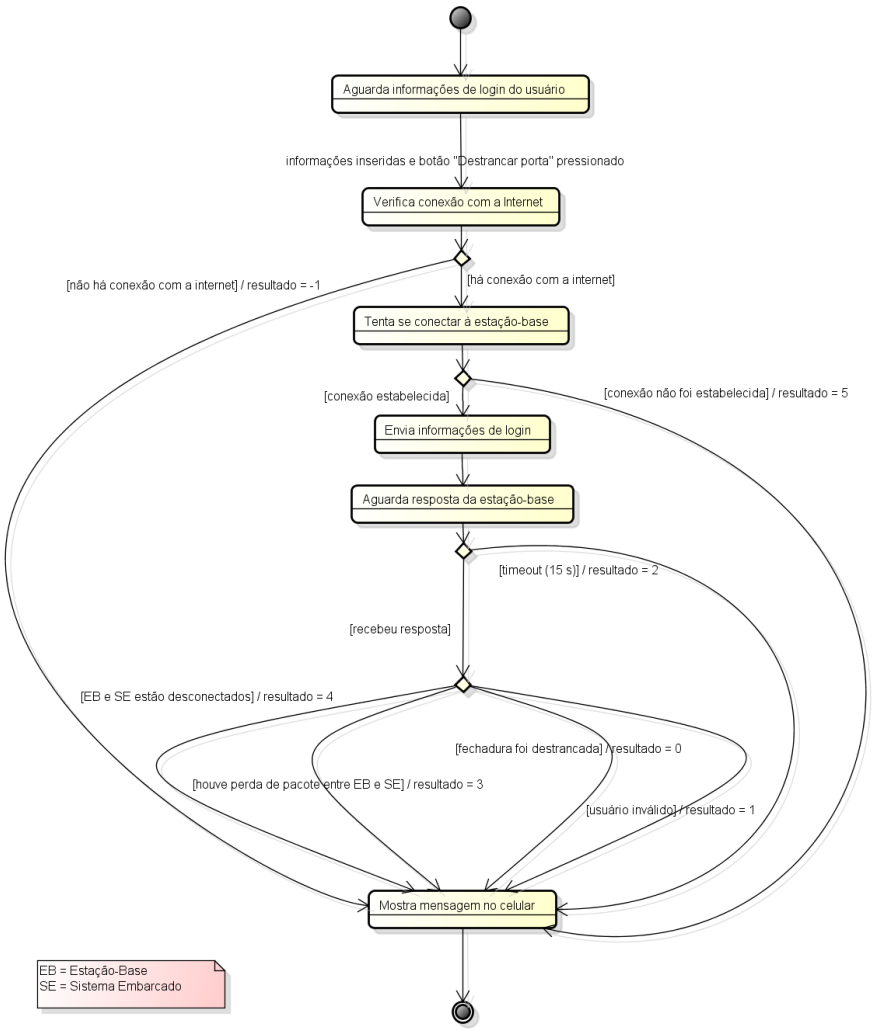
Aguarda informações de login do usuário
informações inseridas e botão "Destrancar porta" pressionado
Verifica conexão com a Internet
[não há conexão com a internet] / resultado = -1
[há conexão com a internet]
Tenta se conectar à estação-base
[conexão estabelecida]
[conexão não foi estabelecida] / resultado = 5
Envia informações de login
Aguarda resposta da estação-base
[timeout (15 s)] / resultado = 2
[recebeu resposta]
[EB e SE estão desconectados] / resultado = 4
[fechadura foi destrancada] / resultado = 0
[houve perda de pacote entre EB e SE] / resultado = 3
[usuário inválido] / resultado = 1
Mostra mensagem no celular
EB = Estação-Base
SE = Sistema Embarcado

Figura 29 - Máquina de estados do aplicativo móvel.

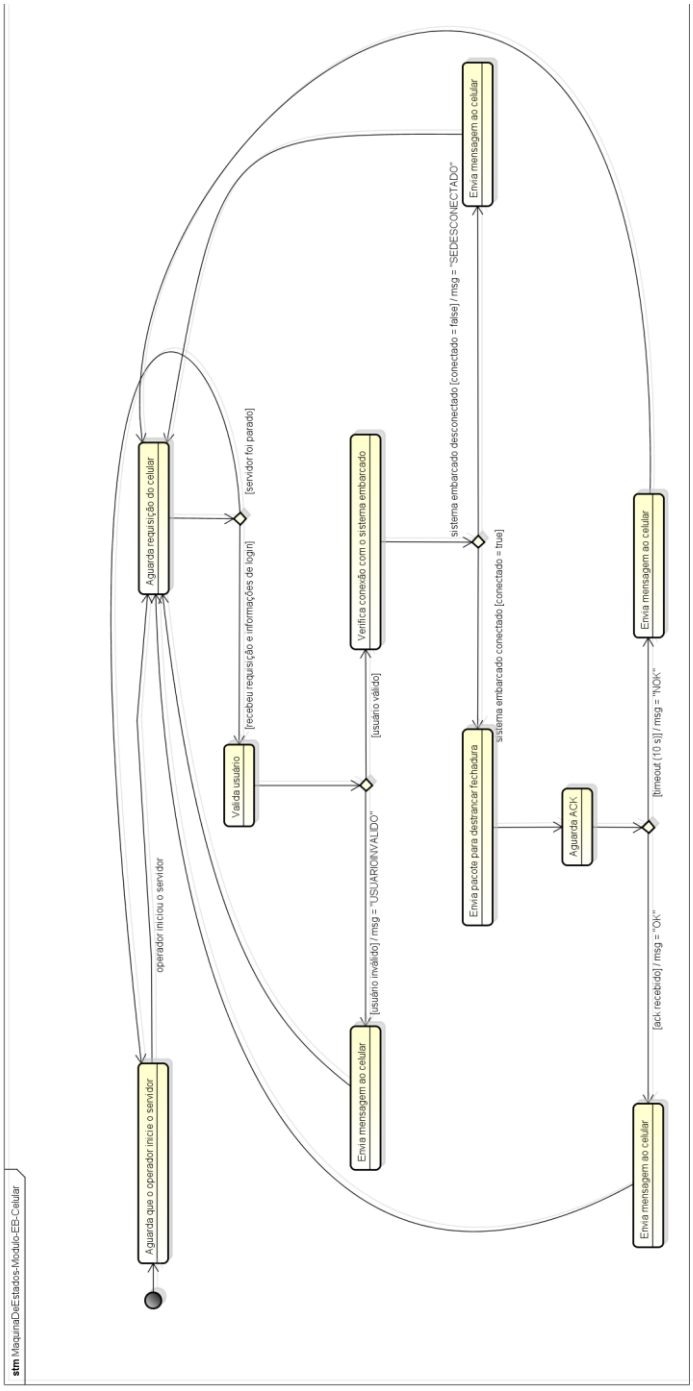
stm MaquinaDeEstados-Modulo-EEB-Celular
Aguarda que o operador inicie o servidor
operador iniciou o servidor
Aguarda requisição do celular
[servidor foi parado]
[recebeu requisição e informações de login]
Valida usuário
[usuário inválido] / msg = "USUARIOINVALIDO"
Envia mensagem ao celular
[usuário válido]
Verifica conexão com o sistema embarcado
sistema embarcado desconectado [conectado = false] / msg = "SEDESCONECTADO"
Envia mensagem ao celular
sistema embarcado conectado [conectado = true]
Envia pacote para destrancar fechadura
Aguarda ACK
[timeout (10 s)] / msg = "NOK"
Envia mensagem ao celular
[ack recebido] / msg = "OK"
Envia mensagem ao celular

Figura 30 - Máquina de estados da estação-base para o recebimento de requisição enviada pelo celular

7.2. COMUNICAÇÃO ENTRE ESTAÇÃO-BASE E SISTEMA EMBARCADO

A comunicação entre a estação-base e o sistema embarcado foi feita utilizando o protocolo TCP, de modo que o sistema embarcado seja o servidor e a estação-base o cliente, podendo enviar minúcias para cadastro, requisições para remoção de digitais, ou envio de uma requisição de abertura via celular.

No sentido inverso, do sistema embarcado para a estação-base, são enviadas informações sobre a ocorrência de um destravamento externo aceito ou rejeitado, ou interno (via botão) para serem armazenados no arquivo de log da estação-base.

Portanto, existem nove mensagens (formatadas em tamanho fixo de 1028 bytes cada) que podem ser trocadas entre a estação-base e o sistema embarcado, mostradas na Tabela 13.

Tabela 13 - Mensagens trocadas entre a estação-base e o sistema embarcado

Mensagem	**Cabeçalho 1 byte**	**1º Parâmetro 3 bytes**	**2º Parâmetro 1024 bytes**
Armazenar digital	0x01h	Id da digital	Minúcias (hexa)
Remover digital	0x02h	Id da digital	X
Notificar entrada	0x03h	Id da digital	X
Notificar saída	0x04h	X	X
Notificar falha de autenticação biométrica	0x05h	X	X
Liberar fechadura	0x06h	X	X
ACK	0x07h	X	X
Terminou sincronização	0x08h	X	X
Heartbeat	0x09h	X	X

Mensagem “Armazenar digital”

Assim que se tenha sido completo um cadastro das 4 digitais na estação-base, as minúcias dessas impressões serão enviadas ao sistema embarcado de forma sequencial (cada digital representa uma mensagem) para armazenamento na memória flash do sensor, onde a cada digital a estação-base esperará o recebimento de uma mensagem de “ACK” do sistema

embarcado antes de enviar a próxima. No caso da ocorrência de uma interrupção na conexão, as minúcias serão reenviadas na próxima sincronização, ativada quando a estação-base se conectar com o sistema embarcado novamente.

Mensagem “Remover digital”

Quando for realizada uma remoção de um usuário na estação-base, será enviada uma mensagem por digital ao sistema embarcado, contendo o ID da digital a ser apagada da memória flash do sensor conectado ao sistema embarcado e a estação-base ficará na espera de um “ACK”. A remoção do usuário e dos arquivos “.bin” das suas digitais associadas só é feita quando todos os 4 “ACKs” forem recebidos, portanto garantindo a consistência de dados. Nota-se que a remoção foi implementada de maneira que ela só pode ser realizada se houver conexão entre o sistema embarcado e a estação-base.

Mensagem “Notificar entrada”

Enviada do sistema embarcado para a estação-base quando ocorrer um destravamento via biometria digital, indicando o ID da digital que realizou tal destravamento, de forma que a estação-base ao receber, envia um “ACK” ao sistema embarcado e associa o ID presente na mensagem com um usuário presente no banco de dados, de forma a mostrar a ocorrência do destravamento no log, com a indicação do horário em que foi recebida essa mensagem e o nome do usuário associado ao ID recebido.

Mensagem “Notificar saída”

Enviada do sistema embarcado à estação-base quando há um destravamento interno via botão. Ao recebimento desta mensagem, a estação-base envia uma mensagem de “ACK” e adiciona o evento de abertura interna via botão ao log.

Mensagem “Notificação de falha de autenticação biométrica”

Enviada à estação-base quando houver uma tentativa de destravamento e a digital analisada pelo sensor biométrico não é reconhecida.

Mensagem “Liberar fechadura”

Enviada ao sistema embarcado quando houver uma requisição de destravamento via celular autenticada pela estação-base.

Mensagem “ACK”

Enviada como resposta à todas as outras mensagens deste protocolo. Sempre que o emissor enviar uma mensagem (que não seja um ACK), este aguarda 10s por uma mensagem “ACK” do receptor. Se o tempo se esgotar e o “ACK” não seja recebido, a transferência é interrompida e o emissor assume que a conexão foi perdida, voltando ao estado desconectado. A conexão necessita ser refeita manualmente pela estação-base.

Mensagem “Terminou sincronização”

Enviada pela estação-base para indicar que a sincronização dos dados foi finalizada. O sistema embarcado só poderá enviar notificações de entrada, saída ou falha após a sincronização dos dados.

Mensagem “Heartbeat”

A mensagem "Heartbeat" é enviada tanto pela estação-base quanto pelo sistema embarcado e tem o objetivo de detectar o estado de conexão.

O sistema embarcado envia um pacote de heartbeat à estação-base se ficar aguardando a chegada de um pacote por mais de cinco segundos. Após o envio, aguarda o ACK e, caso não receba, fecha a conexão atual e volta ao estado de espera por conexões da estação-base.

Similarmente, a estação-base envia um pacote de heartbeat ao sistema embarcado caso fique aguardando a chegada de um pacote por mais de dez segundos. Após o envio, aguarda o ACK e, caso não receba, fecha a conexão. A conexão ao sistema embarcado deverá ser feita pelo operador através da aba de "Configurações" do software da estação-base.

A Figura 31 mostra a máquina de estados da estação-base para o envio de uma mensagem ao sistema embarcado. A Figura 32 apresenta a máquina de estados da estação base na situação de recebimento de uma mensagem do sistema embarcado. Já a Figura 33 mostra a máquina de estados do sistema embarcado ao receber uma mensagem da estação base. Para uma melhor visualização, recomenda-se que estas figuras sejam consultada no CD-ROM que acompanha esta monografia.

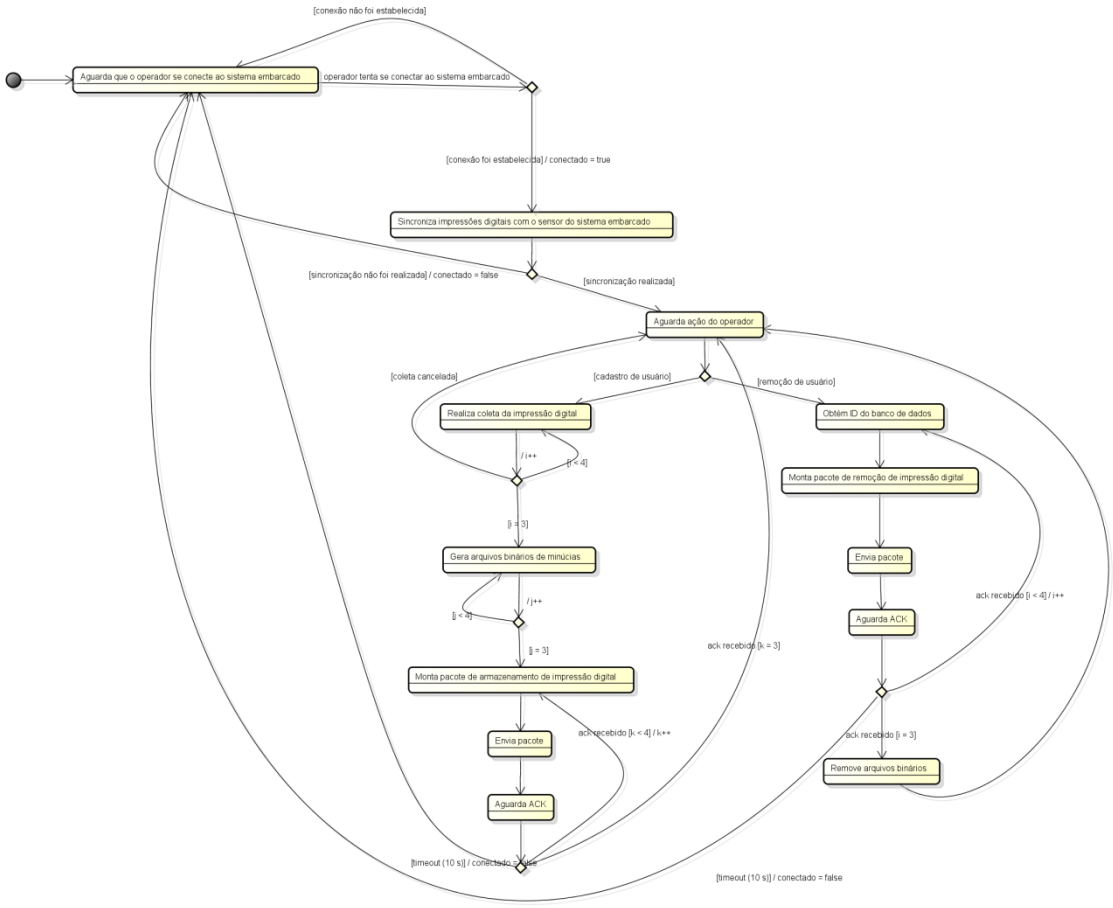
[conexão não foi estabelecida]
Aguarda que o operador se conecte ao sistema embarcado
operador tenta se conectar ao sistema embarcado
[conexão foi estabelecida] / conectado = true
Sincroniza impressões digitais com o sensor do sistema embarcado
[sincronização não foi realizada] / conectado = false
[sincronização realizada]
Aguarda ação do operador
[coleta cancelada]
[cadastro de usuário]
[remoção de usuário]
Realiza coleta da impressão digital
Obtém ID do banco de dados
/ i++
[i < 4]
[i = 3]
Monta pacote de remoção de impressão digital
Gera arquivos binários de minúcias
Envia pacote
/ j++
[j < 4]
Aguarda ACK
ack recebido [i < 4] / i++
[j = 3]
ack recebido [k = 3]
Monta pacote de armazenamento de impressão digital
ack recebido [k < 4] / k++
Envia pacote
ack recebido [i = 3]
Aguarda ACK
Remove arquivos binários
[timeout (10 s)] / conectado = false
[timeout (10 s)] / conectado = false

Figura 31 - Máquina de estados da estação base para um envio de mensagem ao sistema embarcado

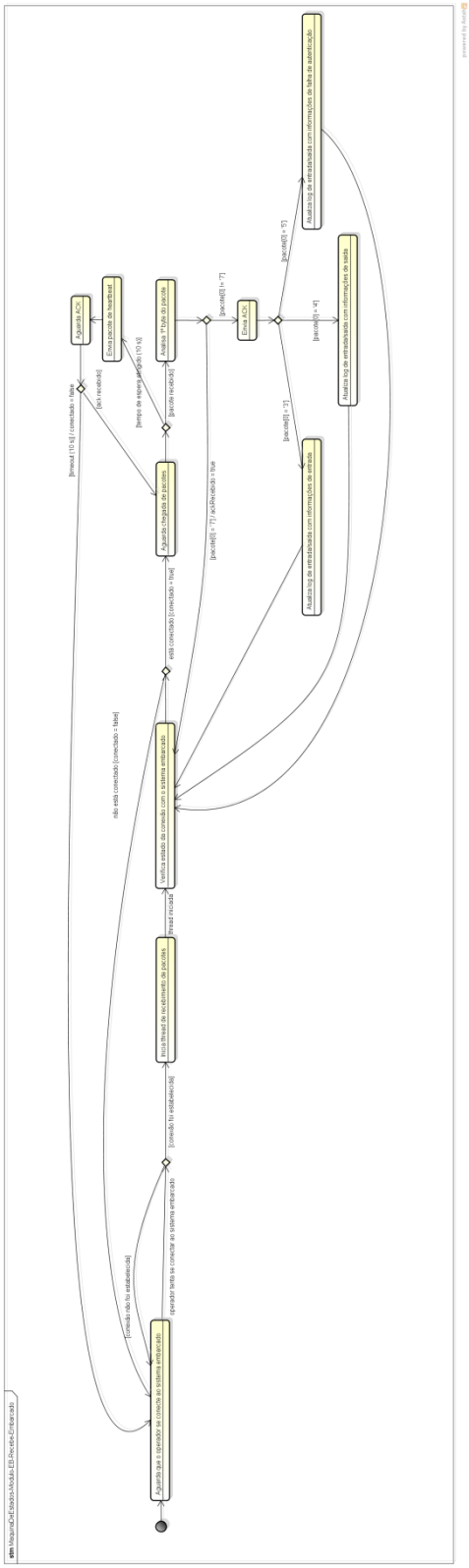

Figura 32 - Máquina de estados da estação base para o recebimento de mensagens do sistema embarcado

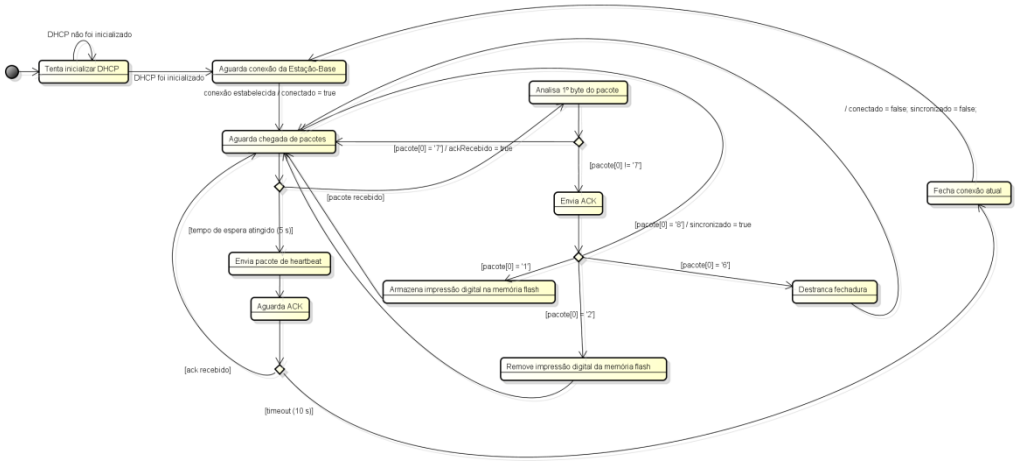
DHCP não foi inicializado
Tenta inicializar DHCP
DHCP foi inicializado
Aguarda conexão da Estação-Base
conexão estabelecida / conectado = true
Aguarda chegada de pacotes
[pacote[0] = '7'] / ackRecebido = true
Analisa 1º byte do pacote
[pacote[0] != '7']
[pacote recebido]
Envia ACK;
[pacote[0] = '8'] / sincronizado = true
[tempo de espera atingido (5 s)]
Envia pacote de heartbeat
[pacote[0] = '1']
[pacote[0] = '6']
Armazena impressão digital na memória flash
Destranca fechadura
Aguarda ACK
[pacote[0] = '2']
Remove impressão digital da memória flash
[ack recebido]
[timeout (10 s)]
/ conectado = false; sincronizado = false;
Fecha conexão atual

Figura 33 - Máquina de estados do sistema embarcado ao receber uma mensagem da estação-base

8. RESULTADOS E CONCLUSÕES

8.1. ANÁLISE DO DESENVOLVIMENTO

Durante o desenvolvimento do projeto notou-se a importância do planejamento e cronograma bem-feitos. Seguindo o que foi planejado e não permitindo que atrasos deslocassem demais o cronograma, a equipe concluiu o projeto e a documentação em dia.

Houve a ocorrência de três dos dez riscos previstos, mas como havia um planejamento de riscos, nenhum deles afetou significativamente o trabalho da equipe. O planejamento e desenvolvimento do hardware tomou um pouco mais de tempo do que o previsto, levando a ocorrência de um atraso de aproximadamente cinco dias, que foi compensado com um aumento da carga de trabalho e compensado em dois dias. Houve também um problema mecânico com a trava, mas que foi facilmente solucionado com o apoio do Prof. Heitor S. Lopes.

Erros ocorreram durante a montagem, mas não representaram grande perda de tempo ou de recursos financeiros. Como a equipe tinha uma reserva financeira e apenas componentes de valor baixo foram danificados, esse tipo de falha não trouxe impactos relevantes.

De início, o déficit de conhecimento parecia imenso para um grupo com pouca experiência de hardware e quase nenhuma de firmware. Mas conforme o semestre avançou e foram aparecendo os primeiros artefatos, viu-se que o medo de enfrentar um desafio como o desse projeto era maior do que a falta de conhecimento. Além disso, as demais disciplinas seguiram e trouxeram novos conhecimentos, que complementaram pesquisas e facilitaram a compreensão dos assuntos envolvidos e o desenvolvimento.

A integração da equipe, embora boa desde o início, pareceu melhorar conforme o projeto avançou. Os membros sempre tiveram uma boa comunicação, o que reduziu consideravelmente o tempo que poderia ser perdido com mal-entendidos e informações incompletas. Durante o desenvolvimento todo, cada membro se esforçou e trabalhou corretamente, o que resultou em facilidade ao acoplar os módulos do projeto.

8.2. INTEGRAÇÃO

A integração deste projeto com as demais disciplinas ficou clara desde o início, quando era necessário definir um método de trabalho e requisitos, aplicando conhecimentos de Engenharia de Software. Disciplinas como Fundamentos de circuitos, e Eletrônica 1 também tiveram seus conhecimentos aplicados no projeto de confecção do hardware, alimentação dos circuitos, posicionamento de componentes básicos, etc.

Para o firmware, muitos conhecimentos de Sistemas Microcontrolados foram utilizados, desde o contato básico com um microcontrolador, até sua programação, utilização de pinos como entrada e saída.

Para a comunicação, aplicou-se conceitos de Comunicação de Dados, para formatar as mensagens, definir um cabeçalho padrão.

Na transmissão de dados, também foram lembrados conceitos de Redes de Computadores 1, os "acknowledges" enviados para informar que uma mensagem foi recebida.

Conceitos de sistemas operacionais foram aplicados quando foram utilizados semáforos e um watchdog para monitorar o sistema embarcado.

Conhecimentos de Segurança e Auditoria de Sistemas também foram aplicados, até mesmo na ideia do projeto quando se empregou mais de um meio de autenticação em diferentes casos (via impressão digital: "something you are" e via uma senha enviada pelo celular: "something you know"), além da criptografia dos dados.

Notou-se a necessidade de maior profundidade nos conhecimentos de hardware em geral, funcionamento de um sistema microcontrolado e sobre transmissão de dados.

8.3. TRABALHOS FUTUROS

Como trabalhos futuros poderiam ser adicionados outros meios de autenticação como reconhecimento de íris, utilização de um cartão magnético ou até mesmo um sistema de reconhecimento de voz.

Outra idéia interessante seria adicionar um teclado númerico para digitação de uma senha pessoal que complementa a autenticação biométrica. No cadastro poderiam ser definidas duas senhas: uma para uso normal e outra para uso apenas em uma situação de risco. Como situação de risco considera-se algum caso em que a pessoa que está autenticando está sendo acuada ou forçada a abrir a sala. Essa senha de emergência poderia acionar um alarme silencioso, ou, ao menos, registrar um evento crítico no log.

Outro ponto importante seria o design de uma estrutura para proteger o sistema, de forma que não fique com os circuitos expostos e tenha apenas as partes que exigem contato com o usuário ou conexões entre módulos expostas.

Poderia também ser feita uma expansão do servidor para que funcione como um servidor central para várias travas em portas diferentes. Os cadastros de usuários poderiam ser feitos em diferentes clientes, mas os logs seriam salvos nesse servidor centralizado, aumentando a segurança.

9. REFERÊNCIAS

[1] COELHO, J. F.; MAIA, V. D. Projeto de uma trava elétrica microprocessada. Disponível em: <http://www.dombosco.fag.edu.br/coor/coopex/5ecci/Trabalhos/Engenharias/Comunicacao/ELETR%D4NICA%20INDUSTRIAL,%20SISTEMAS%20E%20CONTROLES%20ELETR%D4NICOS/687.pdf>. Acesso em: 36 fevereiro 2013.

[2] NUNES, H. C., IBIAPINA, K. O., SANTOS NETO, E. (2011). Sistema de controle do fluxo de funcionários utilizando leitor biométrico. *Engenharia De Computação Em Revista, 1*. Disponível em <http://www3.iesam-pa.edu.br/ojs/index.php/computacao/article/view/472>. Acesso em: 26 fevereiro 2013.

[3] RxTx, disponível em <http://rxtx.qbang.org/wiki/index.php/Main_Page>. Acesso em: 28 de março de 2013.

[4] Data Encryption Standard. Disponível em <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>. Acesso em: 26 de abril de 2013.

[5] Rapid Prototyping for Microcontrollers – mbed. Disponível em: <http://mbed.org/>. Acesso em 26 de abril de 2013.

[6] WatchDog Timer – cookbook. Disponível em: <http://mbed.org/cookbook/WatchDog-Timer>. Acesso em 03 de maio de 2013.

APÊNDICE A - CRONOGRAMAS PLANEJADO E EXECUTADO

	Cronograma planejado		**Cronograma executado**	
Nome	**Início planejado**	**Término planejado**	**Início executado**	**Término executado**
Aplicativo celular	**28/02/2013**	**06/03/2013**	**28/02/2013**	**06/03/2013**
Estudo sobre desenvolvimento para Android	28/02/2013	01/03/2013	28/02/2013	01/03/2013
Diagramas UML	01/03/2013	05/03/2013	01/03/2013	05/03/2013
Protótipo da interface	06/03/2013	06/03/2013	06/03/2013	06/03/2013
Software da estação-base	**28/02/2013**	**06/03/2013**	**28/02/2013**	**06/03/2013**
Diagramas UML	28/02/2013	04/03/2013	28/02/2013	04/03/2013
Protótipo da interface	05/03/2013	05/03/2013	05/03/2013	05/03/2013
Manual do usuário	05/03/2013	06/03/2013	05/03/2013	06/03/2013
Banco de dados	**06/03/2013**	**08/03/2013**	**06/03/2013**	**08/03/2013**
Modelagem do cadastro de usuários	06/03/2013	07/03/2013	06/03/2013	07/03/2013
Modelagem do *log* de entrada-saída	08/03/2013	08/03/2013	08/03/2013	08/03/2013
Comunicação estação-base <-> celular	**07/03/2013**	**07/03/2013**	**07/03/2013**	**07/03/2013**
Estudo sobre *sockets*	07/03/2013	07/03/2013	07/03/2013	07/03/2013
Comunicação estação-base <-> sistema embarcado	**08/03/2013**	**08/03/2013**	**08/03/2013**	**08/03/2013**
Estudo sobre Ethernet	08/03/2013	08/03/2013	08/03/2013	08/03/2013
Hardware	**28/02/2013**	**07/03/2013**	**28/02/2013**	**07/03/2013**
Diagrama em blocos	28/02/2013	04/03/2013	28/02/2013	04/03/2013
Diagrama elétrico	05/03/2013	06/03/2013	05/03/2013	06/03/2013
Lista de componentes	07/03/2013	07/03/2013	07/03/2013	07/03/2013
Deliverable 1	**13/03/2013**	**13/03/2013**	**13/03/2013**	**13/03/2013**
Software da estação-base	**13/03/2013**	**22/03/2013**	**13/03/2013**	**22/03/2013**
Diagramas UML	13/03/2013	15/03/2013	13/03/2013	15/03/2013
Desenvolvimento	18/03/2013	20/03/2013	18/03/2013	20/03/2013
Manual do usuário	21/03/2013	22/03/2013	21/03/2013	22/03/2013
Banco de dados	**20/03/2013**	**26/03/2013**	**20/03/2013**	**26/03/2013**

Implementação da biblioteca de modelagem do log	20/03/2013	20/03/2013	20/03/2013	20/03/2013
Teste de log com dados fictícios	21/03/2013	25/03/2013	21/03/2013	25/03/2013
Alteração de código	26/03/2013	26/03/2013	26/03/2013	26/03/2013
Comunicação estação-base <-> celular	**20/03/2013**	**26/03/2013**	**20/03/2013**	**26/03/2013**
Diagrama de sequência	20/03/2013	20/03/2013	20/03/2013	20/03/2013
Implementação da biblioteca de comunicação	21/03/2013	25/03/2013	21/03/2013	25/03/2013
Teste de comunicação de bits	26/03/2013	26/03/2013	26/03/2013	26/03/2013
Comunicação estação-base <-> sistema embarcado	**20/03/2013**	**27/03/2013**	**20/03/2013**	**27/03/2013**
Diagrama de sequência	20/03/2013	21/03/2013	20/03/2013	21/03/2013
Implementação da biblioteca de comunicação	21/03/2013	26/03/2013	21/03/2013	26/03/2013
Teste de comunicação de bits	26/03/2013	27/03/2013	26/03/2013	27/03/2013
Hardware	**22/03/2013**	**27/03/2013**	**22/03/2013**	**27/03/2013**
Teste em protoboard	22/03/2013	25/03/2013	22/03/2013	25/03/2013
Criação da PCB	26/03/2013	27/03/2013	26/03/2013	29/03/2013
Firmware	**13/03/2013**	**26/03/2013**	**13/03/2013**	**29/03/2013**
Estudo sobre desenvolvimento	13/03/2013	14/03/2013	13/03/2013	14/03/2013
Diagramas UML	15/03/2013	18/03/2013	15/03/2013	29/03/2013
Máquinas de estado	19/03/2013	21/03/2013	19/03/2013	29/03/2013
Diagrama de fluxo de dados	22/03/2013	25/03/2013	22/03/2013	25/03/2013
Deliverable 2	**27/03/2013**	**27/03/2013**	**27/03/2013**	**27/03/2013**
Aplicativo celular	**27/03/2013**	**02/04/2013**	**27/03/2013**	**02/04/2013**
Testes	27/03/2013	27/03/2013	27/03/2013	27/03/2013
Manual do desenvolvedor	28/03/2013	02/04/2013	28/03/2013	10/04/2013
Software da Estação-base	**27/03/2013**	**04/04/2013**	**27/03/2013**	**04/04/2013**
Desenvolvimento	27/03/2013	01/04/2013	27/03/2013	01/04/2013
Manual do desenvolvedor	02/04/2013	04/04/2013	02/04/2013	10/04/2013
Banco de dados	**27/03/2013**	**04/04/2013**	**27/03/2013**	**04/04/2013**
Integração com o software	27/03/2013	28/03/2013	27/03/2013	28/03/2013
Teste com dados reais do	01/04/2013	01/04/2013	01/04/2013	01/04/2013

sistema				
Manual do desenvolvedor	02/04/2013	04/04/2013	02/04/2013	10/04/2013
Comunicação estação-base <-> celular	**05/04/2013**	**09/04/2013**	**05/04/2013**	**09/04/2013**
Testes de comunicação com dados reais de projeto	05/04/2013	05/04/2013	05/04/2013	05/04/2013
Alteração de código	08/04/2013	09/04/2013	08/04/2013	09/04/2013
Comunicação estação-base <-> sistema embarcado	**05/04/2013**	**09/04/2013**	**05/04/2013**	**09/04/2013**
Testes de comunicação com dados reais de projeto	05/04/2013	05/04/2013	05/04/2013	05/04/2013
Alteração de código	08/04/2013	09/04/2013	08/04/2013	09/04/2013
Hardware	**08/04/2013**	**09/04/2013**	**08/04/2013**	**09/04/2013**
Mapa de conexão da PCB com demais módulos	08/04/2013	09/04/2013	08/04/2013	09/04/2013
Firmware	**27/03/2013**	**09/04/2013**	**27/03/2013**	**09/04/2013**
Desenvolvimento	27/03/2013	05/04/2013	27/03/2013	05/04/2013
Simulação virtual	08/04/2013	09/04/2013	08/04/2013	09/04/2013
Deliverable 3	**10/04/2013**	**10/04/2013**	**10/04/2013**	**10/04/2013**
Aplicativo celular	**10/04/2013**	**11/04/2013**	**10/04/2013**	**26/04/2013**
Teste funcional	10/04/2013	11/04/2013	10/04/2013	26/04/2013
Software da estação-base	**12/04/2013**	**15/04/2013**	**12/04/2013**	**15/04/2013**
Teste funcional	12/04/2013	15/04/2013	12/04/2013	26/04/2013
Comunicação estação-base <-> celular	**16/04/2013**	**18/04/2013**	**16/04/2013**	**03/05/2013**
Manual do desenvolvedor	16/04/2013	18/04/2013	16/04/2013	03/05/2013
Comunicação estação-base <-> sistema embarcado	**19/04/2013**	**22/04/2013**	**19/04/2013**	**03/05/2013**
Manual do desenvolvedor	19/04/2013	22/04/2013	19/04/2013	03/05/2013
Hardware	**19/04/2013**	**22/04/2013**	**19/04/2013**	**03/05/2013**
Guia de montagem	19/04/2013	22/04/2013	19/04/2013	03/05/2013
Firmware	**10/04/2013**	**18/04/2013**	**10/04/2013**	**26/04/2013**
Programação do processador	10/04/2013	18/04/2013	10/04/2013	26/04/2013
Deliverable 4	**24/04/2013**	**24/04/2013**	**24/04/2013**	**24/04/2013**
Teste final	25/04/2013	29/04/2013	25/04/2013	03/05/2013
Manual do desenvolvedor	25/04/2013	02/05/2013	25/04/2013	02/05/2013

Finalização do projeto	**03/05/13**	**07/05/13**	**03/05/13**	**07/05/13**
Defesa	**15/05/13**	**15/05/13**		

APÊNDICE B - TABELAS DO PLANO DE RESPOSTAS A RISCOS

Projeto: Trava com Abertura Biométrica ou Remota

1ª Etapa: **Identificação do Risco**

Denominação do risco: Redução permanente na equipe causada pela desistência/trancamento da disciplina ou falecimento de um integrante.	Nº Identificação 1
Descrição do risco: Algum membro da equipe pode desistir/trancar a disciplina a qualquer momento com ou sem aviso prévio ou vir a falecer.	

2ª Etapa: **Avaliação do Risco**

Impacto: □**5(alto)** □4(médio/alto) □3(médio) □2(médio/baixo) □1(baixo) Explique: Um integrante a menos é algo que afeta a equipe toda em aspectos de divisão de tarefas, tempo de execução, divisão de recursos e financeiramente.
Probabilidade: □5(alta) □4(média/alta) □**3(média)** □2(média/baixa) □1 (baixa) Explique: Imprevistos e problemas pessoais são os principais deflagradores de uma desistência/trancamento. E uma morte, normalmente, não pode ser prevista.

3ª Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, responsáveis e datas de conclusão	
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: Os integrantes devem manter o espírito de grupo, comunicando-se com frequência para evitar qualquer imprevisto. No caso de uma desistência, ela deve ser comunicada o mais cedo possível para que seja feito um replanejamento. Devem ser evitados locais, situações e qualquer fator que ponha em risco a vida dos integrantes. Mitigar: As tarefas serão realocadas aos membros que continuarem o projeto, o tempo de dedicação terá que ser estendido, os recursos sofrerão nova divisão.	
Impacto reavaliado: 4(médio/alto)	**Probabilidade reavaliada:** 3(média)

Elaborado por: Gionatta M. Mocellin Laura Wobeto Thayse M. Solis Waldir Marin Neto	Data: 22/01/2013	□ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Projeto: Trava com Abertura Biométrica ou Remota

1ª Etapa: **Identificação do Risco**

Denominação do risco: Perda do software por falha no dispositivo de armazenamento	Nº Identificação 2
Descrição do risco: O software desenvolvido pode ser perdido caso haja alguma falha/dano na memória em que está armazenado.	

2ª Etapa: **Avaliação do Risco**

Impacto: □5(alto) □**4(médio/alto)** □3(médio) □2(médio/baixo) □1(baixo) Explique: A perda do software desenvolvido implica em refazer toda a programação necessária. Isso pode levar algumas horas ou dias, dependendo do tamanho do programa.
Probabilidade: □5(alta) □4(média/alta) □**3(média)** □2(média/baixa) □1 (baixa) Explique: Não há como prever se um dispositivo de memória irá falhar. Isso pode acontecer a qualquer momento, inclusive horas antes da entrega do projeto.

3ª Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, responsáveis e datas de conclusão	
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: Fazer back-up de tudo que for feito, de cada versão e preferencialmente em diferentes dispositivos como pendrives, HDs internos, armazenamento na nuvem, CDs, DVDs, entre outros. Mitigar: A única forma de resolver o problema da perda de um programa sem back-up é fazê-lo novamente. No caso de haver back-up, basta restaurar o arquivo desejado.	
Impacto reavaliado: 1(baixo)	**Probabilidade reavaliada:** 1(baixa)

Elaborado por: Gionatta M. Mocellin Laura Wobeto Thayse M. Solis Waldir Marin Neto	Data: 22/01/2013	□ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Projeto: Trava com Abertura Biométrica ou Remota

1ª Etapa: **Identificação do Risco**

Denominação do risco: Subestimar a complexidade de implementação do software ou hardware	Nº Identificação 3
Descrição do risco: Algumas etapas do projeto podem parecer à primeira vista mais simples do que realmente são.	

2ª Etapa: **Avaliação do Risco**

Impacto: □5(alto) □**4(médio/alto)** □3(médio) □2(médio/baixo) □1(baixo) Explique: Considerar uma tarefa mais simples do que ela é, pode levar a atrasos na implementação.
Probabilidade: □5(alta) □**4(média/alta)** □3(média) □2(média/baixa) □1 (baixa) Explique: Devido à inexperiência em determinadas áreas, é possível que algumas tarefas tenham sua complexidade subestimada.

3ª Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, responsáveis e datas de conclusão	
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: Sempre que possível, conversar com quem tem mais experiência, procurar pesquisar e planejar antes de começar a fazer algo novo. Mitigar: Buscar ajuda com quem tem mais experiência, juntar a equipe para focar na tarefa complexa, mas sem interromper o desenvolvimento do restante do projeto.	
Impacto reavaliado: 3(médio)	**Probabilidade reavaliada:** 3(média)

Elaborado por: Gionatta M. Mocellin Laura Wobeto Thayse M. Solis Waldir Marin Neto	Data: 22/01/2013	□ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Projeto: Trava com Abertura Biométrica ou Remota

1ª Etapa: **Identificação do Risco**

Denominação do risco: Problemas com o microcontrolador, sensor biométrico e/ou demais componentes	Nº Identificação 4
Descrição do risco: Danos causados por sobre-tensão, choques mecânicos, corrompimento do sistema operacional no kit do microcontrolador ou qualquer outra causa que o inutilize total ou parcialmente o hardware utilizado	

2ª Etapa: **Avaliação do Risco**

Impacto: □5(alto) □**4(médio/alto)** □3(médio) □2(médio/baixo) □1(baixo) Explique: O uso incorreto ou a falta de cuidado no manuseio dos componentes pode causar sua perda total ou parcial, mas a equipe possui pelo menos um item reserva dos componentes mais caros/difíceis de obter.
Probabilidade: □5(alta) □4(média/alta) □**3(média)** □2(média/baixa) □1 (baixa) Explique: Manuseio, transporte e utilização serão feitos cuidadosamente.

3ª Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, responsáveis e datas de conclusão
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: Sempre manter os componentes/módulos em superfície estável durante seu uso, jamais deixar seus cabos pendurados quando estiverem conectados. Ao serem transportados, o kit do microcontrolador e o sensor biométrico devem ser alocados de forma a prevenir choques mecânicos, preferencialmente em embalagem anti-estática. Conferir o projeto e a montagem antes de alimentar o circuito. Esta sempre será feita por mais de um integrante, havendo assim uma supervisão mútua a fim de evitar possíveis erros. Haverá pelo menos um componente reserva dos itens mais caros/difíceis de obter para o caso de perda. Mitigar: Devem ser verificados quais componentes foram perdidos, e no caso de ser possível a troca, deve-se proceder com o que for necessário. Caso haja perda total, deve ser empregado o componente reserva.

Impacto reavaliado: 3(médio)	**Probabilidade reavaliada:** 2(média/baixa)

Elaborado por: Gionatta M. Mocellin Laura Wobeto Thayse M. Solis Waldir Marin Neto	Data: 22/01/2013	□ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Projeto: Trava com Abertura Biométrica ou Remota

1ª Etapa: **Identificação do Risco**

Denominação do risco: Não cumprimento dos prazos estabelecidos pelo gerente	Nº Identificação 5
Descrição do risco: Os membros da equipe podem descumprir os prazos estipulados pelo gerente e assim comprometer o andamento do projeto.	

2ª Etapa: **Avaliação do Risco**

Impacto: □5(alto) □**4(médio/alto)** □3(médio) □2(médio/baixo) □1(baixo) Explique: O não cumprimento dos prazos das etapas intermediárias irá interferir no prazo final de entrega do projeto podendo ocasionar sua inviabilização.
Probabilidade: □5(alta) □4(médio/alta) □**3(média)** □2(média/baixa) □1 (baixa) Explique: O gerente acompanhará o andamento das tarefas de forma a orientar a equipe para uma melhor condução dos trabalhos.

3ª Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, responsáveis e datas de conclusão	
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: Deve haver comprometimento da equipe com o projeto, adiantando, sempre que possível, suas tarefas. O gerente deve motivar o grupo para que haja satisfação em cada etapa do trabalho. O planejamento das atividades realizadas em cada semana deve ser detalhado. Os prazos devem ser estabelecidos com alguma folga para evitar constantes atrasos. Mitigar: Deve haver comunicação constante entre os membros da equipe para solucionar o problema de forma prática e satisfatória, evitando atrasos e incentivando a constante troca de ideias entre os membros. No caso de atraso, o tempo de dedicação à tarefa deve ser estendido.	
Impacto reavaliado: 3(médio)	**Probabilidade reavaliada:** 2(média/baixa)

Elaborado por: Gionatta M. Mocellin Laura Wobeto Thayse M. Solis Waldir Marin Neto	Data: 22/01/2013	□ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Projeto: Trava com Abertura Biométrica ou Remota

1ª Etapa: **Identificação do Risco**

Denominação do risco: Redução momentânea da equipe causada por intervenções de ordem física à integridade dos membros	Nº Identificação 6
Descrição do risco: Doenças e acidentes não-fatais podem diminuir temporariamente o número de integrantes da equipe.	

2ª Etapa: **Avaliação do Risco**

Impacto: □5(alto) □**4(médio/alto)** □3(médio) □2(médio/baixo) □1(baixo) Explique: A equipe reduzida terá que aumentar o tempo de dedicação a cada tarefa para suprir a falta de um integrante.
Probabilidade: □5(alta) □4(média/alta) □**3(média)** □2(média/baixa) □1 (baixa) Explique: Doenças e acidentes não são previsíveis, então considera-se uma probabilidade média.

3ª Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, responsáveis e datas de conclusão	
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: Esse tipo de risco é de difícil prevenção, mas indica-se que os membros da equipe tenham cuidado com a saúde e não se exponham a fatores que possam afetar sua integridade física. Mitigar: As tarefas serão realocadas aos demais membros assim como o tempo de dedicação terá que ser estendido, até o retorno do membro que estiver afastado.	
Impacto reavaliado: 3(médio)	**Probabilidade reavaliada:** 3(média)

Elaborado por: Gionatta M. Mocellin Laura Wobeto Thayse M. Solis Waldir Marin Neto	Data: 22/01/2013	□ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Projeto: Trava com Abertura Biométrica ou Remota

1ª Etapa: **Identificação do Risco**

Denominação do risco: Alterações significativas nos requisitos	Nº Identificação 7
Descrição do risco: Após definidos os requisitos, se houverem mudanças, o desempenho da equipe será diretamente afetado.	

2ª Etapa: **Avaliação do Risco**

Impacto: □5(alto) □**4(médio/alto)** □3(médio) □2(médio/baixo) □1(baixo) Explique: Após definidos os requisitos, mudanças nos mesmos podem ocasionar atrasos, dificuldades técnicas e falta de material.
Probabilidade: □5(alta) □4(média/alta) □3(média) □**2(média/baixa)** □1 (baixa) Explique: Não há como prever exatamente qual a probabilidade de uma alteração nos requisitos, mas ela deve ser evitada.

3ª Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, responsáveis e datas de conclusão	
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: Procurar definir cautelosamente cada requisito, refletindo suas consequências em todo projeto. Evitar que sejam incluídos novos requisitos durante a etapa de desenvolvimento, em especial requisitos que aumentem demais a complexidade, exijam muitos mais componentes ou muito tempo de trabalho. Mitigar: Planejar rapidamente no que o novo requisito afeta o projeto, incluir as mudanças e já realocar tempo e pessoas assim como adquirir os recursos necessários.	
Impacto reavaliado: 3(médio)	**Probabilidade reavaliada:** 2(média/baixa)

Elaborado por: Gionatta M. Mocellin Laura Wobeto Thayse M. Solis Waldir Marin Neto	Data: 22/01/2013	□ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Projeto: Trava com Abertura Biométrica ou Remota

1ª Etapa: **Identificação do Risco**

Denominação do risco: Atraso na entrega de componentes importados	Nº Identificação 8
Descrição do risco: Dependendo da origem do componente, esse pode levar meses para ser entregue, ficar retido na alfândega ou mesmo perder-se no caminho até o Brasil.	

2ª Etapa: **Avaliação do Risco**

Impacto: □5(alto) □4(médio/alto) **□3(médio)** □2(médio/baixo) □1(baixo) Explique: O não-recebimento de algum componente importado implica na mudança do projeto e rápido planejamento de novas alternativas. Contudo, a equipe já possui uma unidade do componente mais crítico a ser importado.
Probabilidade: □5(alta) **□4(média/alta)** □3(média) □2(média/baixa) □1 (baixa) Explique: Não há como saber o prazo exato de entrega de um pedido internacional, nem prever uma perda do mesmo.

3ª Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, responsáveis e datas de conclusão	
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: Dar preferência a produtos nacionais ou que estejam disponíveis em estoque brasileiro, evitar compras no exterior com valor igual ou acima de cinquenta dólares americanos (para que não fiquem retidos na aduana). E no caso de componentes que vem do exterior, adquirir pelo menos um a mais de reserva em compras diferentes. Mitigar: Caso haja atraso na entrega de um componente, providenciar uma nova compra do mesmo, pois não se sabe se ele não foi perdido. Adiantar as demais etapas do projeto que não dependam do componente, atentando ao prazo máximo para, se for o caso, buscar uma alternativa que substitua o componente faltante.	
Impacto reavaliado: 2(médio/baixo)	**Probabilidade reavaliada:** 4(média/alta)

Elaborado por: Gionatta M. Mocellin Laura Wobeto Thayse M. Solis Waldir Marin Neto	Data: 22/01/2013	□ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Projeto: Trava com Abertura Biométrica ou Remota

1ª Etapa: **Identificação do Risco**

Denominação do risco: Problemas com a comunicação de dados	Nº Identificação 9
Descrição do risco: Dificuldade de implementar a comunicação, sem a qual a troca de informações entre estação-base e sistema embarcado e estação-base e celular é inviabilizada.	

2ª Etapa: **Avaliação do Risco**

Impacto: □5(alto) □4(médio/alto) **□3(médio)** □2(médio/baixo) □1(baixo) Explique: Sem comunicação o sistema fica inoperante.
Probabilidade: □5(alta) □4(média/alta) **□3(média)** □2(média/baixa) □1 (baixa) Explique: A comunicação exige conhecimento de protocolos de comunicação e de tecnologias necessárias à sua implementação.

3ª Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, responsáveis e datas de conclusão	
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: Deve ser realizado um plano backup antes que ocorra qualquer problema. Estudar cuidadosamente as tecnologias a serem utilizadas assim como os protocolos necessários para a comunicação entre a estação-base e o sistema embarcado e entre a estação-base e o celular. Mitigar: Migração para outras tecnologias e protocolos viáveis, já estudados anteriormente.	
Impacto reavaliado: 3(médio)	**Probabilidade reavaliada:** 2(média/baixa)

Elaborado por: Gionatta M. Mocellin Laura Wobeto Thayse M. Solis Waldir Marin Neto	Data: 22/01/2013	□ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Projeto: Trava com Abertura Biométrica ou Remota

1ª Etapa: **Identificação do Risco**

Denominação do risco: Roubo ou extravio parcial ou total do hardware	Nº Identificação 10
Descrição do risco: A qualquer momento um componente pode ser perdido, furtado ou ser levado à força por um agente externo.	

2ª Etapa: **Avaliação do Risco**

Impacto: □5(alto) □4(médio/alto) **□3(médio)** □2(médio/baixo) □1(baixo) Explique: A perda de um componente, por qualquer meio, exige a substituição do mesmo por um igual ou equivalente.
Probabilidade: □5(alta) □4(média/alta) **□3(média)** □2(média/baixa) □1 (baixa) Explique: Os componentes serão bem armazenados, levando em consideração o fato que a preferência em um possível assalto sejam celulares e carteiras.

3ª Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, responsáveis e datas de conclusão	
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: Guardar os componentes cuidadosamente, em um lugar estipulado, para evitar que se esqueça onde ele está e seja considerado perdido. O(s) componente(s) backup devem ser guardados em locais diferentes dos componentes que estão em uso, para evitar que um acidente danifique todos eles. No caso de mais de um componente back-up, eles devem ser conservados em locais distintos. Evitar trafegar com os componentes por zonas de criminalidade, onde o índice de assaltos seja grande. Mitigar: Providenciar rapidamente um novo componente ou usar os reservas.	
Impacto reavaliado: 2(médio/baixo)	**Probabilidade reavaliada:** 3(média)

Elaborado por: Gionatta M. Mocellin Laura Wobeto Thayse M. Solis Waldir Marin Neto	Data: 22/01/2013	□ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

APÊNDICE C - MÁQUINA DE ESTADOS COMPLETA DA ESTAÇÃO BASE

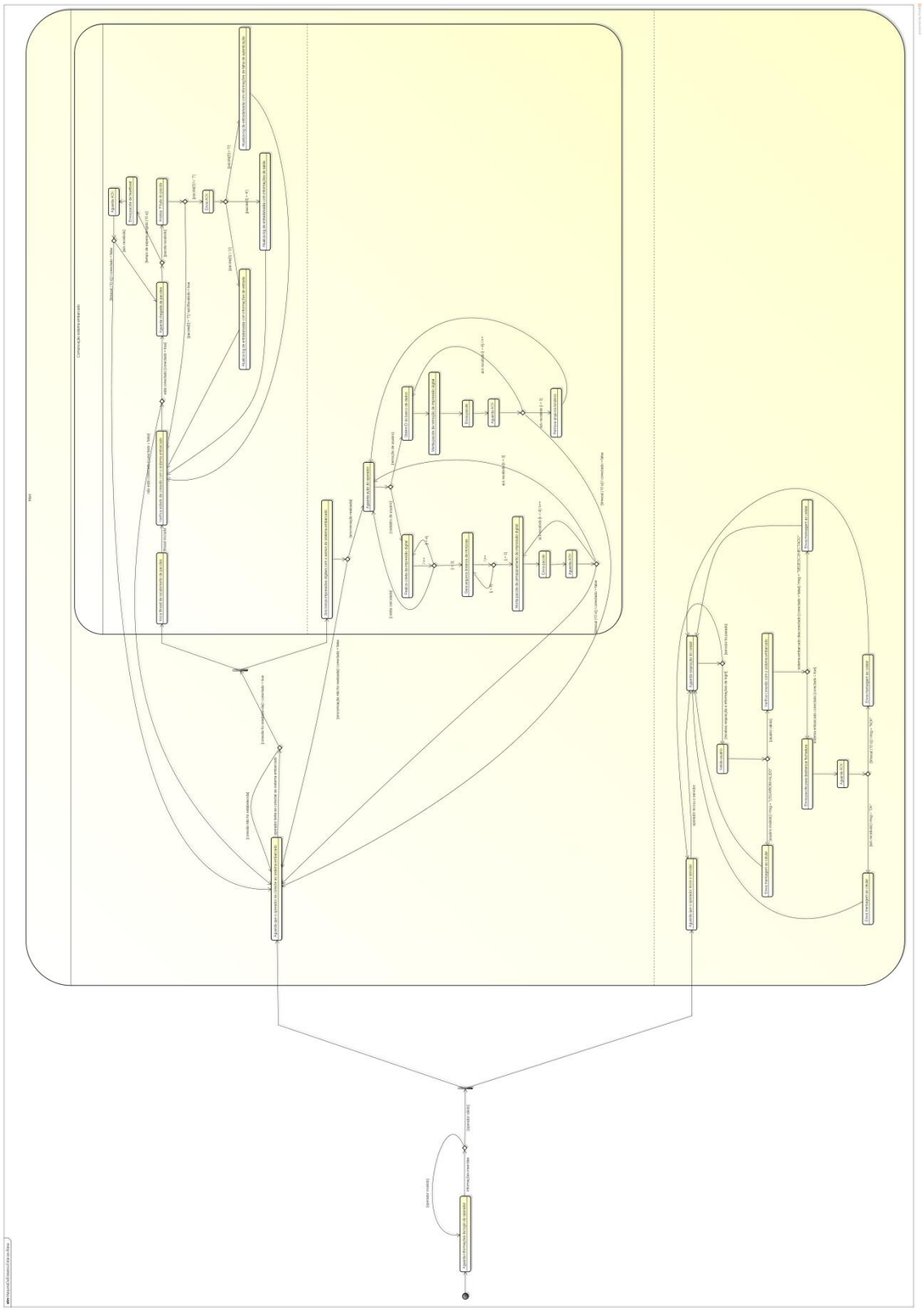

Figura 34 - Máquina de estados completa da estação base

APÊNDICE D - MÁQUINA DE ESTADOS COMPLETA DO FIRMWARE

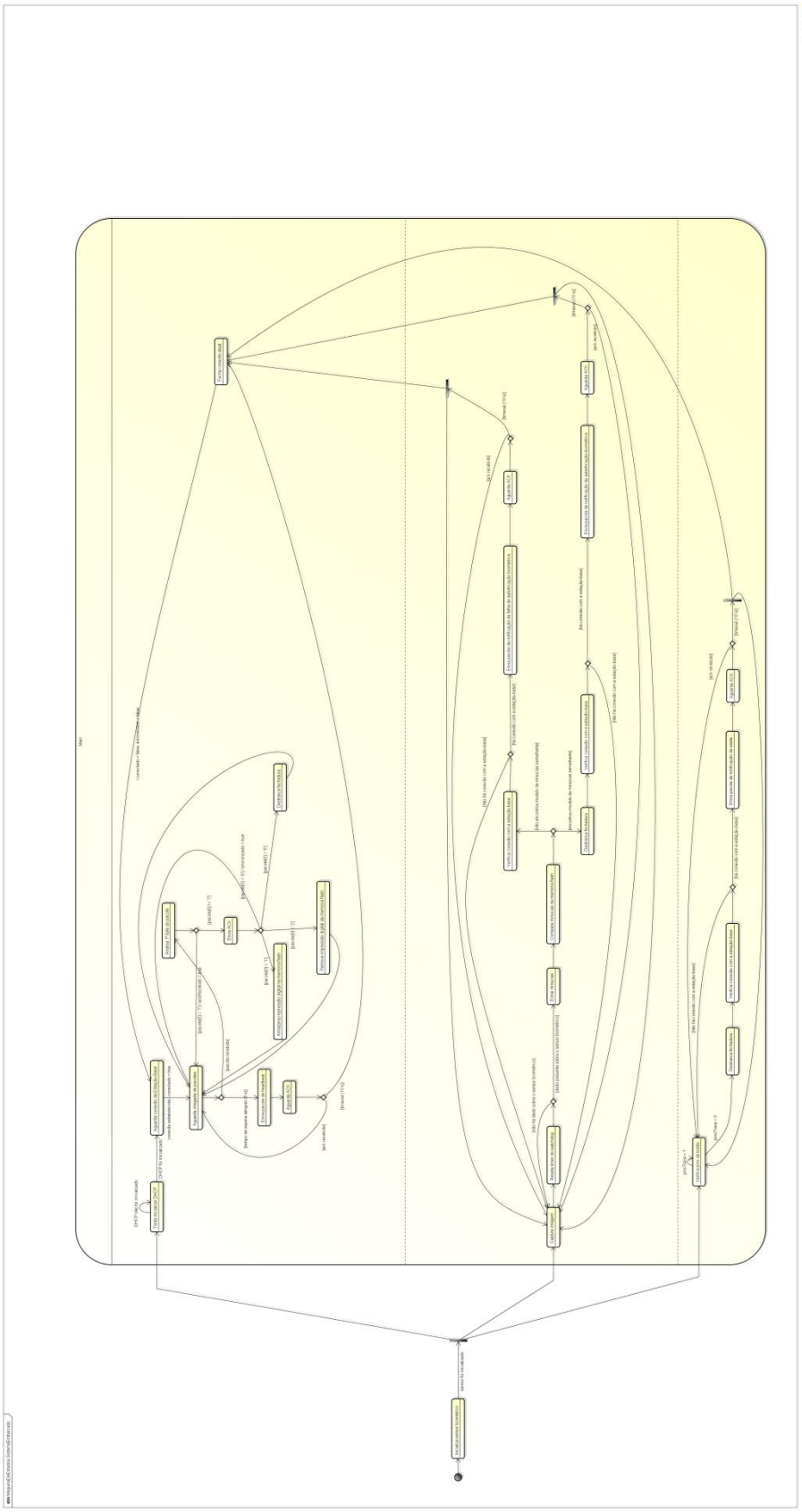

Figura 35 - Máquina de estados completa do sistema embarcado